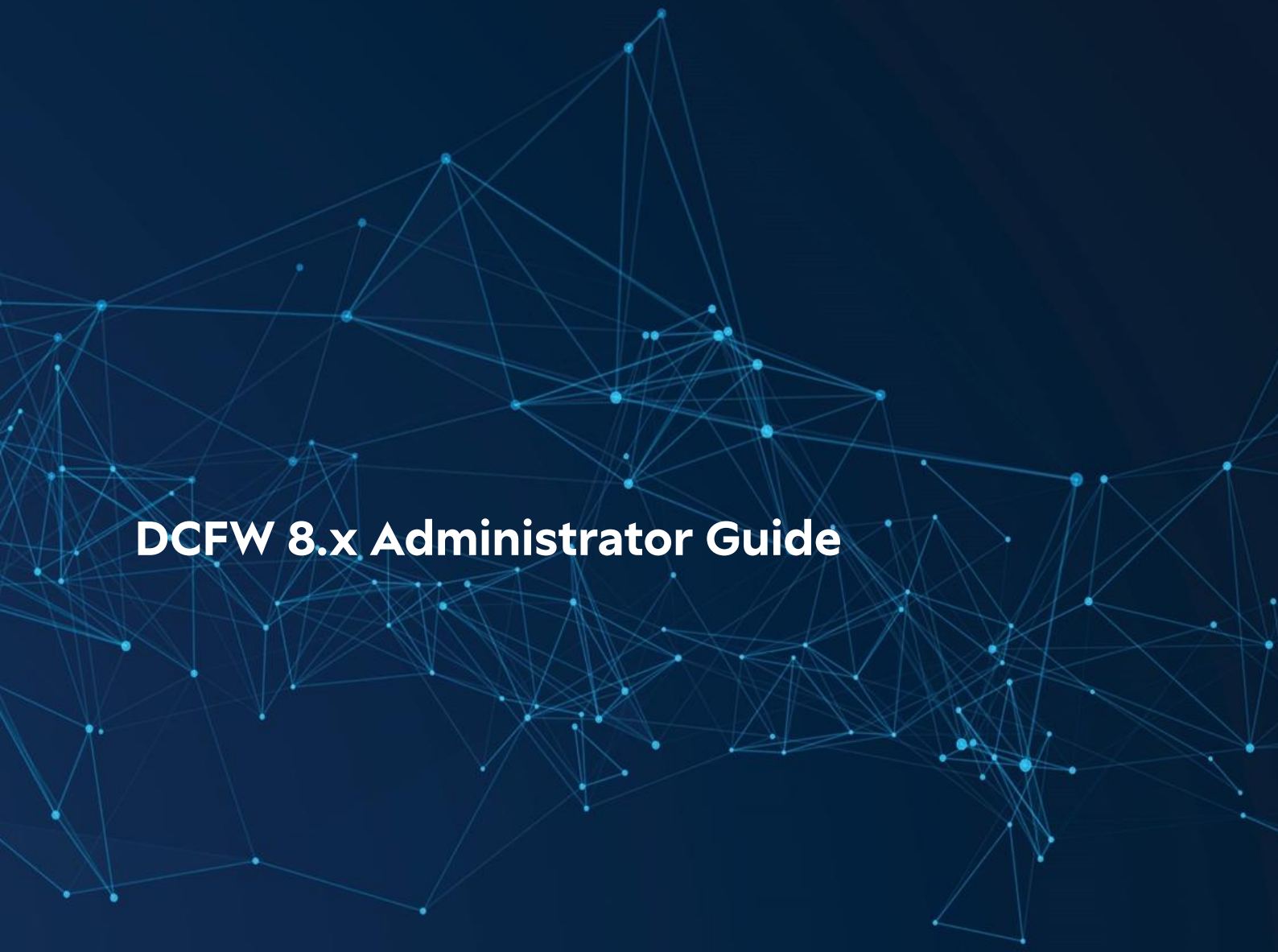
UserGate



DCFW 8.x Administrator Guide

Table of Contents

INTRODUCTION

NETWORK SECURITY AND THREAT PROTECTION

Firewall

UserGate DCFW is a high-performance next-generation firewall designed to protect large corporate networks and data centers. DCFW filters traffic that passes through certain protocols (such as TCP, UDP, or IP), thereby protecting the network from hacker attacks and various intrusions that are based on exploiting these protocols.

Intrusion Detection and Prevention

The intrusion detection and prevention system (IDPS) enables malicious activity within the network to be identified. It focuses on real-time threat detection, logging, and prevention, as well as reporting.

The administrator can also create custom IDPS signatures aimed to protect specific services and include them into IDPS profiles along with UserGate-supplied signatures. IDPS profiles are integrated into firewall rules. When signatures from such a profile are encountered, the action configured for the signatures will be taken, and a corresponding entry will be made in the IDPS Log.

DoS and Network Flood Protection

DCFW allows you to configure network flood protection parameters per network zone for TCP (SYN-flood), UDP, and ICMP protocols by setting an alert threshold (the number of requests from a single IP address that triggers logging) and a packet drop threshold (the number of requests, after which the packets are dropped with a corresponding log entry).

It is possible to configure exceptions — e.g., for zones that send a large number of UDP packets due to VoIP use.

Scenario-Based Security Policy Configuration

With DCFW, the attack detection to response time can be reduced considerably thanks to a scenario-based mechanism called SOAR, Security Orchestration, Automation and Response.

Currently enjoying strong popularity, this concept enables the administrator to define scheduled or detection-triggered scenarios that specify automated actions to be taken in response to various events. Such an approach enables flexible configuration of security policies, reduces human intervention by automating repetitive tasks, and allows prioritizing of scenarios to ensure rapid response to critical threats.

INTERNET PERFORMANCE AND RELIABILITY IMPROVEMENT

Clustering and High Availability Support

UserGate DCFW supports two types of clusters: the configuration cluster that can provide unified configuration to cluster nodes and the high-availability (HA) cluster aimed at ensuring fail-safe network operation. The HA cluster has two modes of operation: Active-Active and Active-Passive. Both modes support user session synchronization, which provides user-transparent traffic switching between nodes.

FTP over HTTP

The FTP over HTTP module allows access to content on an FTP server from a user's browser.

Multi-Provider Support

If your system is connected to several providers, UserGate DCFW allows you to configure a separate gateway for each of them to provide Internet access. The administrator can also configure traffic load balancing between the providers by specifying a weight for each gateway or defining one of the gateways as the main so that the system switches to other providers if the main gateway is unavailable.

Traffic Shaping Management

Traffic shaping rules are used to limit the bandwidth for certain users, hosts, services, or applications.

WCCP Support

WCCP support allows to use DCFW within infrastructures with WCCP servers, such as Cisco routers.

TRAFFIC MANAGEMENT AND INTERNET ACCESS CONTROL

Traffic Routing and Resource Publishing

With DCFW, you can use both static and dynamic routing. Dynamic routing is carried out using the OSPF and BGP protocols, making it possible to employ DCFW in corporate networks with complex routing.

The administrator can create NAT rules in the system (to provide Internet access to users), as well as rules for secure publishing internal resources to the Internet using DNAT.

User Authentication and Authorization

The platform supports different user authentication mechanisms, such as Captive portal, Kerberos, NTLM, etc. The user accounts can originate from a variety of sources, including LDAP, Active Directory, FreeIPA, TACACS+, RADIUS, and SAML IDP. SAML IDP, Kerberos, and NTLM allow for transparent (i.e., without requesting a username and password) authentication of Active Directory domain users on the UserGate device. The Captive portal also supports user authentication with certificates that use public key infrastructure (PKI).

Thanks to the UserID feature, transparent user authentication is possible on selected UserGate devices. Active Directory logs, syslog and RADIUS accounting messages are used as authentication data sources for that purpose. To do this, the UserID agent sends requests to AD servers via the WMI protocol. While in the syslog scenario, it listens on the syslog port and collects information sent by the syslog servers; and in the RADIUS scenario, it receives RADIUS accounting messages from NAS servers. Then the information is filtered by user login / logout events. Based on the obtained data, it searches for the user in the user catalogs of the log source. If the user is found, the user's authorization data is sent to all UserGate devices specified in the source redistribution profile, and the user is logged in to DCFW.

The administrator can configure security rules, link bandwidth, firewall rules, as well as content filtering and application control rules for individual users, user groups, or all known or unknown users. In addition, UserGate supports the application of security rules to terminal service users via dedicated Terminal Services Agents and the use of an authorization agent for Windows platforms.

For better user account security, multi-factor authentication with TOTP (Time-based One Time Password Algorithm) tokens, SMS, or email should be used.

LOGS AND REPORTS

Logs and Reports (Description)

The platform enables real-time system monitoring using event, web access, IDPS, and traffic logs. The administrator can configure automatic log export to SSH, FTP, and Syslog servers for more convenient analysis. Reports allow administrators to provide different slices of data about security events, configurations, or user actions.

Reports can be created automatically according to previously created rules and templates and sent to recipients by email.

VIRTUAL NETWORKS

VPN

A VPN (Virtual Private Network) is used to set up virtual logical networks that operate on top of other networks, such as the Internet. UserGate DCFW supports two types of VPN networks — a Remote Access VPN (client / server model), and a Site-to-Site VPN (server / server model).

To create secure tunnels, L2TP/IPsec, IPsec (IKEv1), and IPsec (IKEv2) protocols are used. UserGate has its own VPN client, UserGate Client, and also supports working with standard clients for the majority of popular operating systems, including Windows, Linux, Mac OS X, iOS, Android, and others

OTHER FEATURES

Load Balancing

DCFW supports load balancing for various services within the local network. Load balancing can be provided for internal servers published in the Internet (DNAT), and for internal servers without publishing.

DNS Filtering

With DCFW, you can configure the operation of DNS servers and set up a DNS proxy service that enables users' DNS requests to be intercepted and modified according to the administrator's needs. The platform also enables you to filter user DNS requests.

Using Notifications

UserGate DCFW supports monitoring using the SNMP v2c and SNMP v3 protocols. Both query-based (SNMP queries) and notification-based control (SNMP traps) are supported.

In addition, you can create notification profiles to alert users on certain events using the SMTP (email) and SMPP (SMS) protocols.

Role-based administrator access to UserGate DCFW controls

By default, the system has one super-administrator who is allowed to create other administrator accounts and grant them permissions to view and edit various sections.

As an additional security measure for console access, you can turn on certificate-based authorization for administrators.

Interface Types

UserGate DCFW allows you to add and configure tagged VLAN interfaces, as well as combine several physical interfaces into a single aggregated logical interface (bond) using LACP (Link Aggregation Control Protocol) to increase the bandwidth or improve link availability.

You can combine the interfaces to the bridge to filter the traffic at the level 2 without changing the network infrastructure of the company. If the bridge is created in the HSC DCFW using the network card supporting the bypass mode, you can combine two interfaces into the bypass bridge. You can find the detailed information in the Configuring interfaces section.

LICENSING

DCFW Licensing

Basic license

UserGate DCFW licensing is based on platform performance parameters and depends on:

- type of hardware platform (for hardware and software systems);
- the number of supported virtual machine cores (for a virtual image).

If you try to register invalid hardware with a key with performance limitation, an error will appear: Entered PIN code is licensed for another type of UserGate device, or configuration of this server is not licensed, for example, number of actual CPU cores exceeds the number of CPU cores licensed.

> **Note**
> **If a virtual machine is registered with the valid key and additional cores are added in the future, only the number of cores allowed by the license will be active in the virtual machine.**

The basic product license is perpetual (software and library updates are not included).

Additionally Licensed Modules

The following modules can be additionally licensed.

Module	Description
Security Updates (SU)	The SU module grants the right to receive updates of: • The UserGate software • Intrusion detection system signatures • L7 Application signatures The module is supplied as an annual subscription. After one year, you will need to renew the license to continue receiving updates
Cluster	The module includes a license to allow UserGate devices to operate in cluster mode. The license term is unlimited.

License Activation Procedures

Online Activation

During online activation, the UserGate device/software accesses the licensing server https://reg2.usergate.com. Technical details is sent to the server, including the UserGate software version number, PIN code, product name, device model, etc. The response is the license term and the list of modules permitted by the license.

If any modules that were previously present in the system are not on this list, they are deactivated and their license is revoked. Newly added modules are activated.

After that, the UserGate device checks the license once a day. If everything is OK, the device operates normally. If the license check is successful, this event is recorded in the logs.

If the licensing servers are unavailable, 14 connection attempts are made at 120 second intervals. If unsuccessful, the attempts are stopped for 24 hours, followed by 14 more attempts to connect to the activation server again. If the license fails to connect to the activation server during the license validity period, the license is blocked upon expiration (modules with expired license stop working). Each activation server connection error is recorded in the logs.

Online Activation Procedure

To register the device:

1. In the device admin web console, go to the **Dashboards** section,

2. In the **License** widget, click **No license**, enter the PIN code and register the device.

If the node is in a closed perimeter without direct access to the Internet, you can activate or update the license through a proxy server. To do this, select the **Use a proxy server for activation and updates** mode. Then specify the IP address and port of the upstream proxy server. If necessary, specify the login and password for authentication on the proxy server.

Offline Activation

Offline activation of licenses is required for UserGate devices located in an isolated network without Internet access and without the ability to activate via a proxy server.

The offline licensing process includes the following steps:

1. Request generation: creation of a request file for offline activation on the licensed device.

2. Request activation: processing the generated request file using the offline PIN code activation service.

3. Applying the license: downloading the activated file back to the licensed device.

Request generation

To generate a request file for offline license activation:

1. Access the licensed device using a web browser at the following address: https://<IP-address>:8001?features=offline-reg .

IP address is the IP address of the licensed device.

2. In the device web console, go to the **Dashboards** section.

3. In the **License** widget, click **No license**.

4. In the device activation window, click **Begin offline activation**.

5. Enter your device PIN and download the generated request file for offline activation.

Request activation

From a computer with Internet access, contact the offline activation service (to enter the service, you will need authorization in the Unified authorization center) and activate the generated request file.

Applying the license

Upload the activated file to the licensed device. To do that:

1. In the **Dashboards** section of the licensed device, in the **License** widget, open the offline activation window.

2. Select **Finish offline activation**.

3. Specify the activated file received from the offline activation service.

The licensing process is complete.

For more info on the offline license activation procedure, see the Offline License Activation section.

INITIAL CONFIGURATION

General Information

UserGate DCFW is available as a hardware and software system (HSC) or as a virtual machine image (virtual appliance) designed to be deployed in a virtual environment. As a virtual appliance, DCFW is supplied with ten Ethernet interfaces. In the form of an HSC, it can have 2 to 64 Ethernet ports.

Virtual Appliance Deployment

UserGate DCFW Virtual Appliance is a quick way to deploy a VM with pre-configured components. The VM image is supplied in the OVF format (Open Virtualization Format) supported by VMWare, Oracle VirtualBox, and Qcow2 platforms for QEMU-KVM virtualization systems. For Microsoft Hyper-V, a VM disk image is supplied.

Note

For the correct operation of the VM, 32 GB RAM and 4-core virtual CPU are recommended as a minimum. Your hypervisor must support 64-bit operating systems.

Note

For the internal database to function correctly, the x86 architecture SSE4.2 micro-instruction set must be supported by the virtual environment processors. Any processor based on the x86 architecture released after 2008 must support SSE4.2.

Working With a Virtual Image

To get started with the virtual appliance, follow these steps:

1. Download the latest version of the virtual appliance from the official UserGate website.

2. Import the VM image into your virtualization system. Instructions on how to import a VM image can be found on the VirtualBox and VMWare websites. For Microsoft Hyper-V, you first need to create a VM, specify the downloaded VM image as the disk, and then disable Integration Services in the settings for the newly created VM.

3. Configure the VM parameters. Increase the size of the RAM for the VM. In the VM properties, set a minimum of 8GB and add 1GB for each 100 users.

4. Create an additional disk of the required size.

The default disk size is 100GB, which is usually not enough to store all logs and settings. In the VM properties, increase the disk size. Recommended size: 1 TB.

For QEMU-KVM, the default system partition size is 8GB. At the first boot, the system will automatically detect the additional disk and expand its system partitions.

This command example adds a 100 GB disk to a QEMU-KVM system:

```
qemu-img create -f qcow2 -o
preallocation=metadata,refcount_bits=16,lazy_refcounts=on,cluster_size=
4K your-disk-name.qcow2 100G
```

5. Start the UserGate VM. During loading, **Factory reset** is performed. UserGate uses this step to configure network adapters and increase the partition size on the hard disk to the full size specified at Step 4.

UserGate NGFW is supplied with four interfaces assigned to zones:

- **Management**: the first VM interface;
- **Trusted**: the second VM interface;
- **Untrusted**: the third VM interface;
- **DMZ**: the fourth VM interface.

> **Note**
>
> **If network interfaces on a DCFW virtual machine have been deleted by means of the hypervisor, they will be marked in the web interface as deleted using the strike-through icon.**

> **Note**
> **When a UserGate virtual machine is being cloned using vSphere, then the source's virtual machine MAC addresses must be deleted in the VMX file containing the parameters of the cloned VM.**

Optimizing Virtio-Based Network Interface Performance

To optimize the performance of virtio-based network interfaces in KVM, oVirt, and zVirt virtualization environments, we recommend enabling Multi Queues mode on the hypervisor and setting 8 queues per network interface.

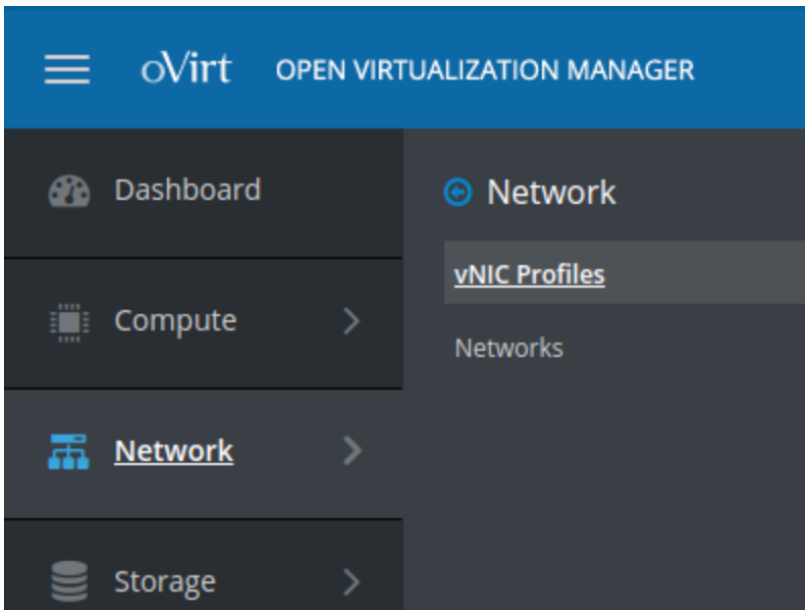
For example, for the oVirt platform (please refer to the oVirt documentation), you need to connect to the hypervisor's CLI and enter the following command to set up 8 queues for vNIC:

```
engine-config -s "CustomDeviceProperties={type=interface;prop={other-nic-properties;queues=[1-9][0-9]*}}"
```

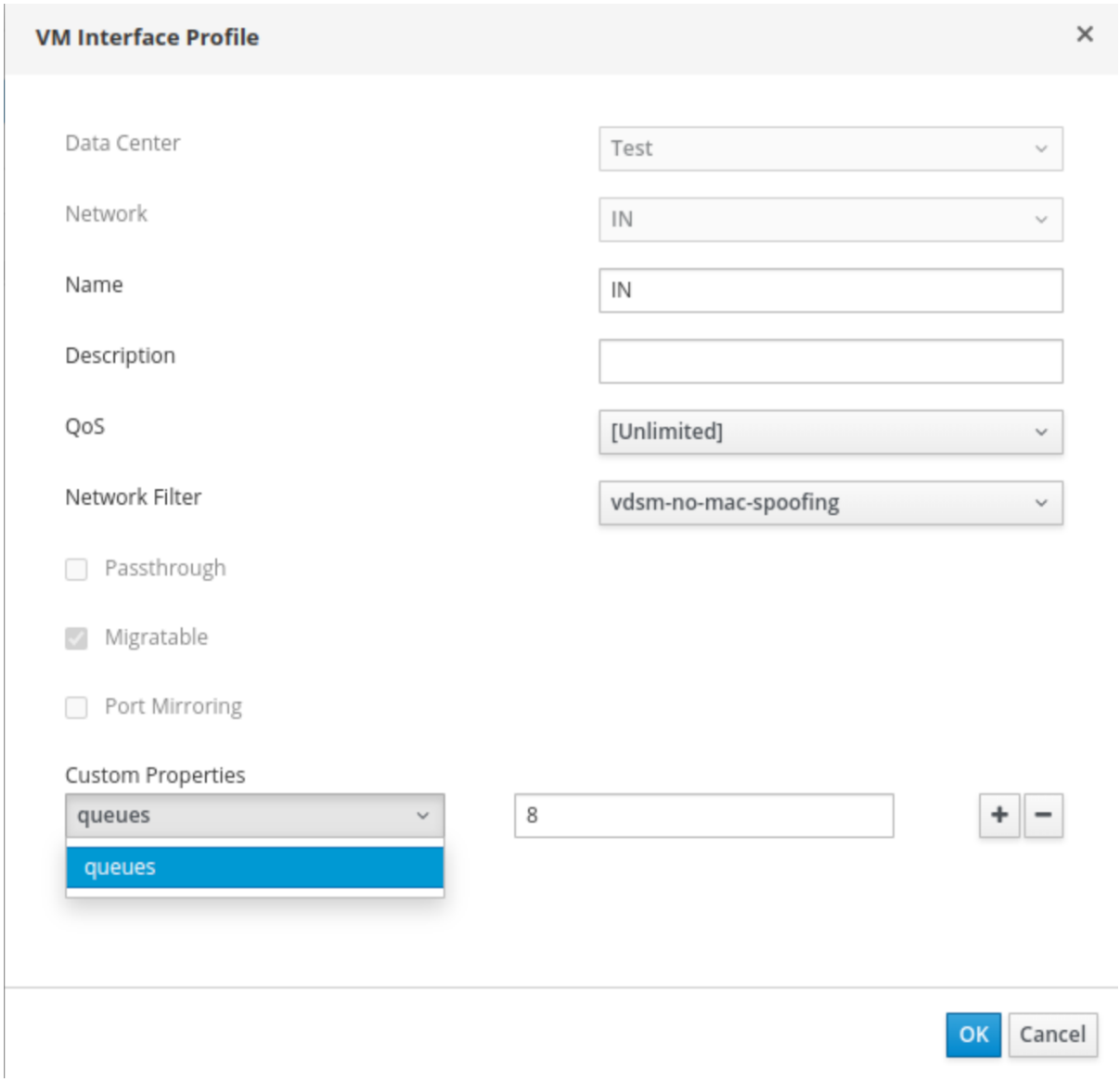
Instead of using the `other-nic-properties` parameter, you'll have to add a list of existing customized rules (if any). You can check whether such rules exist using the command:

```
engine-config -g "CustomDeviceProperties"
```

After entering the command, you need to access the vNIC profiles in the administrator portal:



Select to edit the network card profile assigned to DCFW, select **queues** in the **Custom Properties** drop-down list, and then specify the required number of queues:



Automate UserGate DCFW Deployment Using Cloud-init

Cloud-init is an industry standard for cross-platform VM instance initialization in cloud services of different providers. UserGate DCFW supports initial configuration using the Cloud-init mechanism. The firewall setup is done using two modules:

- Setup using CLI (file with a #utm-config header). All CLI commands can be used for full instance setup.
- License activation (file with a #utm-license header).

No other cloud-init modules are supported.

Example configuration file with CLI commands (user-data):

```
#utm-config
#set password for initial Administrator (Admin). Obligatory comand.
password 123
#Set addresses and settings for network interfaces:
set network interface adapter port1 \
ip-addresses [ 172.16.6.9/24 ] \
enabled on \
zone "Trusted"
set network interface adapter port2 \
ip-addresses [ 172.16.8.9/24 ] \
enabled on \
zone "Untrusted"
set network interface adapter port3 \
ip-addresses [ 172.16.7.9/24 ] \
enabled on \
zone "DMZ"
#Create network gateway to Internet:
create network gateway \
ip 172.16.8.2 \
default on \
interface port2 \
virtual-router default \
enabled on
#Create firewall rule to allow traffic from Trusted to untrusted
security zones:
create network-policy firewall \
position 1 upl-rule ALLOW \
src.zone = Trusted \
dst.zone = Untrusted \
enabled(true) \
name("Cloud-Init: Allow from Trusted to Untrusted")
```

marks the beginning of a comment, and a backslash (\) denotes a wrap to the next line.

All CLI commands available to the administrator can be used in this file. For more details on CLI commands, see the Command Line Interface (CLI) section.

You can activate the instance being created by specifying the licensing parameters in a separate file. Note that activation is only possible if the instance has Internet access. Example license activation file (vendor-data):

```
#utm-license
pin_code: UGN4-XXXX-YYYY-ZZZZ-AAAA
reg_name: UG-test
email:  email@company.com
user_name: John
last_name: Doe
company: UserGate
country: UAE
region: Dubai
```

The two files can be merged into one using the multipart format:

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0
--//
Content-Type: text/utm-config; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="config.txt"
#utm-config
password 123
set network interface adapter port1 \
ip-addresses [ 172.16.6.9/24 ] \
enabled on \
zone "Trusted"
set network interface adapter port2 \
ip-addresses [ 172.16.8.9/24 ] \
enabled on \
zone "Untrusted"
set network interface adapter port3 \
ip-addresses [ 172.16.7.9/24 ] \
enabled on \
zone "DMZ"
create network gateway \
```

```
ip 172.16.8.2 \
default on \
interface port2 \
virtual-router default \
enabled on
create network-policy firewall \
position 1 upl-rule ALLOW \
src.zone = Trusted \
dst.zone = Untrusted \
enabled(true) \
name("Cloud-Init: Allow from Trusted to Untrusted")
--//
Content-Type: text/utm-license; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="license.txt"
#utm-license
pin_code: UGN4-XXXX-YYYY-ZZZZ-AAAA
reg_name: UG-test
email:  email@company.com
user_name: John
last_name: Doe
company: UserGate
country: UAE
region: Dubai
--//
```

Settings can be transferred to DCFW:

- Via a cloud provider. For example, with Digital Ocean, when creating a virtual machine (droplet), settings must be entered into the optional **User data** field (**Select additional options → User data**). Other cloud services providers support similar methods of settings transfer.
- Using a mounted .iso image. The disk must contain files named `meta-data`, `user-data`, and `vendor-data` with the following contents:
    - meta-data: `instance-id: vm1`
    - user-data — with CLI instance setup commands:

```
#utm-config
#set password for initial Administrator (Admin). Obligatory
comand.
password 123
#Set addresses and settings for network interfaces:
set network interface adapter port1 \
ip-addresses [ 172.16.6.9/24 ] \
enabled on \
zone "Trusted"
...
```

- vendor-data — with optional licensing information:

```
#utm-license
pin_code: UGN4-XXXX-YYYY-ZZZZ-AAAA
reg_name: UG-test
email: email@company.com
...
```

To create an ISO disc on Linux, you can use the following utility:

```
mkisofs -joliet -rock -volid "cidata" -output nocloud.iso meta-data
user-data vendor-data
```

Mount the resulting ISO disk on the UserGate VM. After the first successful boot, the VM will receive all settings specified for it in the created files.

Network Environment Requirements

For the correct operation of UserGate DCFW, it must have access to the following Internet servers:

- registration server — reg2.usergate.com (TCP ports 80, 443);
- UserGate software update server: updates.usergate.com, (TCP ports 80, 443).

When creating a configuration cluster, the following protocols must be allowed between the nodes:

- Settings replication: TCP ports 4369, 9000-9100
- Web console service: TCP port 8001

For more on network availability requirements, see the Network Environment Requirements appendix.

Connecting to UserGate DCFW

The port0 interface is configured to receive an IP address automatically from a DHCP server and assigned to the **Management** zone. The initial configuration is done via the administrator's web console connection via the port0 interface.

If it is not possible to assign an IP address to the Management interface automatically using DHCP, it can be set explicitly from the CLI (Command Line Interface). For more details on using CLI, see the Command Line Interface section.

> **Note**
> **If the device has not undergone initial setup, use Admin as the login and usergate as the password for accessing the CLI.**

Other network interfaces are disabled and require further configuration.

For initial setup, follow these steps:

Step	Description
1. Connect to the management interface.	Connect to the device interface: • **When a DHCP Server Is Used**. Connect the port0 interface to the corporate network with a working DHCP server. Enable UserGate DCFW. After downloading, the DCFW console will display the IP address received by the interface. Connect to the device's web console at https://<DCFW_IP_address>:8001 to further activate the product. • **Static IP address**. Start UserGate DCFW. Using the CLI, assign the desired IP address to the port0 interface. Perform initial setup via the CLI or connect to the DCFW web console at that IP address. The address string should look like: https://<DCFW_IP_address>:8001.

Step	Description
	For more details on using CLI, see the Command Line Interface section.
2. Select a language	
3. Set a password	
4. Configure zones, set IP addresses of the network interfaces, and connect UserGate NGFW to the corporate network	In the **General setting → Network → Interfaces** section, enable the desired network interfaces, assign valid IP addresses that correspond to your networks, and bind the interfaces to the respective zones. For more details on network interface management, see the Network Interface Configuration section. The system is supplied with a number of predefined zones: • Management (management network), port0 interface; • Trusted (LAN); • Untrusted (Internet); • DMZ; • Cluster; • VPN for remote access; • VPN for Site-to-Site; • Tunnel inspection zone.
5. Configure the Internet gateway	In the **General setting → Network → Gateways** section, specify the IP address for the Internet gateway on an Internet-connected network interface in the **Untrusted** zone. For more details on configuring gateways, see the Gateway Configuration section.
6. Specify the system DNS servers	In the **General settings → Network → DNS** section, specify the IP addresses of your provider's or corporate DNS servers. For more details on DNS management, see the DNS Configuration section.
7. Set the server time	In the **Settings → UserGate → General settings → Server time settings** section, configure time synchronization with NTP servers.
8. Register NGFW	In the **Dashboard** section, in the **License** widget, click **No license** and enter the PIN code to register the product. To activate the system, Internet access is required. For more details on product licensing, see the Licensing section.
9. Create NAT rules	In **Settings → Network policies → NAT and routing**, create the necessary NAT rules. A NAT rule has already been created for

Step	Description
	Internet access for **Trusted** network users: **NAT from Trusted to Untrusted**. For more details on NAT rules, see the NAT and Routing section.
10. Create firewall rules	In **Settings → Network policies → Firewall**, create the necessary firewall rules. To ensure unrestricted internet access for users on the **Trusted** network, the **Allow trusted to untrusted** rule has already been created; you just need to enable it. For more details on firewall rules, see theFirewall section.

After completing these steps, UserGate DCFW is ready to work. You can also further customize other settings.

Step	Description
Create additional administrators	In the **Settings → UserGate → UserGate Administrators** section, create additional system administrators and grant them the necessary rights (roles).
Configure user authorization	In the **General setting → Users and devices** section, create the required user authorization methods. The easiest option is to create local DCFW users in the **General settings → Users and devices → Users** section with specified IP addresses or use the system without user identification (use the **Any** user in all rules). For other user authorization options, see the Users and Devices section.

DEVICE SETUP

Setting up General Parameters

This section describes the basic parameters of UserGate DCFW, configured in the **General settings → UserGate → Settings** section.

Name	Description
Timezone	The timezone for your location. Used in rule schedules and for the correct display of time and date in reports, logs, etc.

Name	Description
Default interface language	The language to use by default in the console.
Web console authentication mode	The method of authenticating the user (administrator) when logging in to the web management console: • **Login and password**. The administrator must provide their login name and password to get access to the web console. • **X.509 certificate**. For certificate-based authentication, you need a user certificate signed with the certificate of the web console Certification Authority and installed in the browser. When this authentication mode is turned on, the login name and password mode is disabled. You can restore the login name and password authentication mode afterwards using CLI commands. • **User certificate profile.** Authentication Using PKI certificates uses user certificate profile allows managing certificates that provide the security and authentication of network connections.
SSL profile for web console	Select an SSL profile to build a secure web console access link. For more details on SSL profiles, see the SSL Profiles section.
SSL Profile for block/ authorization pages	Select an SSL profile to build a secure link for displaying web resource block pages and the captive portal's auth page. For more details on SSL profiles, see the SSL Profiles section.
Automatic session closure timer (min)	Configure the automatic session termination timer that will expire in case there is no administrator activity in the web console.
Endpoint device SSL profile	Select an SSL profile to create a secure communication link between DCFW and UserGate Client endpoint devices. For more details on SSL profiles, see the SSL Profiles section.

Name	Description
Endpoint device certificate	The certificate that will be used to create a secure communication link between DCFW and UserGate Client endpoint devices. **Attention!** **Endpoint devices remember the certificate, therefore, when it is changed, you need to distribute the root CA certificate to the connected endpoint devices. The certificate must be installed into the local machine's Trusted Root Certification Authorities certificate store.**
Server time settings	Configure the time synchronization settings. • **Use NTP servers**: use the NTP servers from the provided list for time synchronization. • **Primary NTP server**: the primary time server address. Default value: `pool.ntp.org`. • **Secondary NTP server**: the secondary time server address. • **Server time**: allows time setting on the server. The UTC timezone should be used.
Cache settings	You can configure the following proxy server cache settings: • **Caching mode on/off**: enable or disable caching. • **Cache exclusions**: the list of URLs that will not be cached. • **Max cache object size (MB)**: objects larger than this will not be cached. It is recommended to leave the default value of 1MB. • **RAM size (MB)**: the amount of RAM reserved for the cache. This should not be set to more than 20% of the system RAM.
PCAP Configuration	Configure the traffic logging triggered when IPS signatures are encountered. These are the options for packet capture: • **No capture**; • **One packet**; • **Previous packets** (4 to 30 packets); • **Previous and following packets** (previous: 4 to 30; following: 2 to 15).

Name	Description
	Attention! **A large PCAP value can slow down data processing significantly.**
Change tracker	If this option is enabled and **Change types** have been defined, any change to the configuration introduced by the administrator using the web console will require that the administrator specify the change type and a description for the change. Here are some possible examples of change types: • directive; • order; • scheduled maintenance. The number of change types is not limited.
Update center	This is where you configure update downloads for UserGate software (UGOS) and system libraries provided on subscription (URL filtering category database, IDPS, IP/URL/content type lists etc.). **Software updates**: configure the update channel (stable, beta), checking for new UGOS updates and downloading offline updates. After checking, you can manually download and install available updates. During the UGOS update installation process, you can create a device restore point. This will allow you to restore the previous version of UGOS if problems arise. The action will be available in the start menu after installing the UGOS update. **Libraries updates**: check for libraries updates, download updates, and configure the automatic library check and download schedule. You can check for library updates and download the latest updates by clicking the **Check for updates** link. You can configure automatic library updates by clicking the **Con figure** link. Once the license is activated, the following library set will be updated automatically: • L7 application signatures; • UserGate antimalware; • IP list of botnets;

Name	Description
	• IDPS Signatures; • URL list of phishing sites; • compact and full UserGate URL Filtering (URLF) libraries. • Ad Blocking; • Trusted CAs; The list of automatically updated libraries can be modified by the administrator in the automatic update schedule settings. For each library, you can configure a schedule for automatically checking and downloading updates. You can select from the following schedule options: Setting up a schedule for automatic library updates. When setting up a schedule, you can select one of the preset values or enter the time manually using cron format: . For manual entry, you can use the following characters: • (*): all values. For example, in the hour field, the symbol means the backup should run every hour. • (-): range of values. • (,): is used as the delimiter of values. • (/): is used to indicate step between values. If you select the **Apply for all updates** checkbox, the selected library's schedule will be applied to all libraries in the list **Note** **To reduce system load, it is recommended to set automatic updates only for the libraries you use.**
Modules	You can configure the following DCFW modules: • **HTTP(S) proxy port**: allows you to specify a non-standard (alternative) port number that will be used to connect to the built-in proxy. By default, TCP port 8090 is used. If changed, the port continues working. The ports 2200, 8001, 4369, 9000-9100 may not be used, as they are used by the internal DCFW services. • **Captive portal auth domain**: a service domain used by DCFW for user authorization through captive portal. The users need to be able to resolve the domain provided here into the IP address of the UserGate network interface to which they are connected. If the users have the DCFW's IP address specified as the DNS server,

Name	Description
	address resolving is configured automatically. The default name is auth.captive. It can be changed to another domain name used in the organization. • **Captive portal logout domain**: an internal domain used by DCFW users to terminate their sessions (log out). The users need to be able to resolve the domain provided here into the IP address of the DCFW network interface to which they are connected. If the users have the DCFW's IP address specified as the DNS server, address resolving is configured automatically. The default name is logout.captive. It can be changed to another domain name used in the organization. • **Block page domain**: an internal domain used to display a block page to users. The users need to be able to resolve the domain provided here into the IP address of the DCFW network interface to which they are connected. If the users have the DCFW's IP address specified as the DNS server, address resolving is configured automatically. The default name is block.captive. It can be changed to another domain name used in the organization. • **FTP over HTTP**: enable or disable the module that provides access to content on FTP servers from a user browser. The FTP proxy must be specified explicitly in the user browser. The administrator can restrict access to FTP resources using content filtering rules (only the Users and URL criteria are supported). • **FTP over HTTP domain**: an internal domain used to provide FTP over HTTP service to users. The users need to be able to resolve the indicated domain into the IP address of the DCFW interface which they are connected to. If the users have the UserGate server's IP address specified as the DNS server, address resolving is configured automatically. The default name is ftpclient.captive. It can be changed to another domain name used in the organization. • **Password for terminal server agent**: set the password to be used by terminal server authorization agents for connection. • **LLDP settings**: configure the use of the Link Layer Discovery Protocol (LLDP) that enables the network equipment in the local area network to notify devices about its existence, report its characteristics, and receive similar information from the devices. These settings are required: ◦ **Transmit delay**: how long the device will wait before sending advertisements to the neighbors after a change in the LLDP protocol's TLV parameter or the local system state (e.g., a

Name	Description
	changed hostname or management address). Specified in seconds and can take values from 1 to 3600. ◦ **Transmit hold**: the hold multiplier. The transmit delay multiplied by the transmit hold determines the time to live (TTL) for LLDP packets. Can take values from 1 to 100.
Log Analyzer	This section displays information about the log database server. • **Local/Remote server**: A local or external log database server (LogAn). • **Log Analyzer version**: the software version of a log database server. • **Device version**: the device's software version. • **Device ID**: a unique device ID for integration with an external Log Analyzer server. **Attention!** **If an external LogAn is specified, that LogAn server will be processing and exporting logs, generating reports, and handling other statistics.**
UserGate Management Center agent	Here you can configure device connection to the central management console (UGMC) that can be used to manage a UserGate device fleet from a single point. TCP ports 2022 and 9712 are used for connection to the UGMC server. The parameters include: • **Enabled/Disabled**: enable or disable management via UGMC. • **UserGate Management Center address**: server address in IPv4 address format, FQDN (IDN address can also be used). • **Device code**: a token required to connect to UGMC. UGMC can be used as the software and signature update source.

Name	Description
	Note **You will be able to connect UserGate Management Center to DCFW only when the DCFW port is located in the default virtual router. Please see the Virtual routers section for more details.**
Upstream proxy	Configure upstream proxy settings for user traffic redirection. The settings include the proxy type (HTTP(S), SOCKS5), IP address, and port, as well as the login and password for authenticating with the proxy (if required).

Device management

The **Device management** section defines the following DCFW settings:

- Clustering
- Diagnostics settings
- Server operations
- Backup
- Settings export and import

Please see the Clustering and High availability section for more details on clustering.

Diagnostics

This block is designed to configure diagnostic parameters and provide remote access to the UserGate technical support service for the purpose of analysis and troubleshooting.

Diagnostic parameters

Using the **Diagnostic details** parameter, you can set the device logging level. Available levels:

- **Off**: diagnostics logs are disabled
- **Error**: log only device errors
- **Warning**: log only errors and warnings
- **Info**: log only errors, warnings, and additional information
- **Debug**: log all possible events

Logging at levels **Warning**, **Info** and **Debug** may reduce device performance, so it is recommended to set the levels to **Error** or **Off** unless otherwise suggested by UserGate technical support.

Manage logs

You can download the diagnostic logs for sending them to UserGate support. Web console logs and system logs are available for download. Selected logs can be downloaded after archiving them by clicking **Start archiving logs**.

To delete archived (currently inactive) logs, click **Clear log files**.

Remote assistance

To provide access to the device for the UserGate technical support service for diagnostics and troubleshooting purposes, you must activate the remote assistance function and obtain session access parameters.

The process of connecting to the device is as follows:

1. The UserGate device administrator activates the remote assistance function.

2. The device establishes a secure connection to the UserGate remote assistance server via the SSH protocol. If the connection is successful, the UserGate device interface will display the session access parameters: identifier and token.

3. The UserGate device administrator transfers session access parameters to the UserGate technical support specialist.

4. The technical support specialist establishes a secure connection via the SSH protocol with the UserGate remote assistance server and connects to the UserGate device using session access parameters.

Server operations

In this section, you can perform the following server maintenance actions:

Name	Description
Server operations	• **Reboot**: reboot the DCFW. • **Shutdown**: shut the DCFW down.
Upstream proxy settings to check licenses and updates	Configure the upstream HTTP(S) proxy server settings for license and software updates for the DCFW. Specify the IP address and port of the upstream proxy server. If necessary, specify login and password for authentication on the upstream proxy server.

The UserGate team is continuously working to improve its software and provides UserGate product updates as part of a Security Update license module subscription (for more details on licensing, see the chapter Licensing).

System backup management

This section allows you to manage DCFW backup and restore features.

Under **Device management → System backup management**, click **Create backup**. The system will save the current server settings in a file named:

`backup_PRODUCT_NODE-NAME_DATE.gpg, where:`

Where:

- `PRODUCT` is the product type: DCFW, LogAn, or MC;
- `NODE-NAME` is the UserGate node name;
- *DATE* is the date and time when the backup was created as YYYY-MM-DD-HH-MM. The time is in UTC time zone.

To interrupt the backup process, press the **Stop** button. The backup record will be displayed in the device event log.

In the **Device management → System backup management**, click **Restore from backup** and specify the path to the previously created settings file to upload it to the server. Restore will be suggested in the tty console when the device reboots.

In addition, the administrator can configure a scheduled file upload to external servers (FTP, SSH). To create a schedule for uploading settings, follow these steps:

1. In the **Device management → System backup management**, click **Add** and enter a name and description for the rule.

2. Specify the rule name and description in the **General** tab.

3. In the **Remote server** tab, specify the external server settings:

- **Server type**: FTP or SSH
- **Address**: the server's IP address
- **Port**: the server's port
- **Login name**: the user account on the remote server
- **Password/Repeat password**: the password for the user account
- **Directory path**: the path on the server where the settings will be uploaded The path on the server must already exist. The system itself will not create non-existent folders!

If using an SSH server, you can use key authorization. To import or generate a key, select **SSH key setup** and specify **Generate key** or **Import key**.

Important!

If you re-create a key, the existing SSH key will be deleted. The public key must reside on the SSH server in the user keys directory

/home/user/.ssh/

in the

authorized_keys

file.

When initially configuring the SSH backup export rule, connection verification is mandatory (**Check connection** button). When the connection is verified, the fingerprint is placed in known_hosts. The files are not sent without verification.

> **Important!**
> **If you change the SSH server or reinstall it, the backup files will be unavailable, because the fingerprint has changed. This protects you from spoofing.**

3. In the **Schedule** tab of the rule, specify when the settings should be uploaded. To set the time in the crontab format, specify it as follows: (minutes:0-59) (hours:0-23) (days of the month:1-31) (month:1-12) (day of the week:0-6, where 0 is Sunday).

Each of the first five fields can be defined using:

- An asterisk (*) denotes the entire range (from the first number to the last).
- A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7.
- Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23".
- The asterisk and dash are also used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

Exporting and importing settings

The administrator can save the current DCFW settings in a file and later restore them on the same or another DCFW device. This is different from a backup in that importing/exporting the settings does not preserve the current state of all system components — only the current settings are saved.

Settings export is a cluster function. It works as follows: when a rule for settings exporting is created on one of the cluster nodes, it is automatically replicated to the other cluster nodes. The export files themselves are created and sent separately on each node.

> **Note**
> **Importing/exporting the settings does not preserve the cluster state or license information. After completing the import, you will need to re-register DCFW using the existing PIN code and, if necessary, re-create the cluster.**

> **Note**
> **If TOTP-based multifactor authentication is used, TOTP keys are not stored; re-authentication will be required. Re-initialization is required.**

You can export either all settings (except those listed above) or export network settings only. When only the network settings are exported, the following information is preserved:

- DNS settings
- DHCP Configuration
- The settings for all interfaces, including tunnels
- Gateway settings
- Virtual router (VRF) settings
- WCCP Configuration
- Zone settings.

Under **Device management → Settings export and import**, click **Export → Export all settings** or **Export network settings**. The system will save the current server settings in a file named:

`utm-utmcore@nodename_version-YYYYMMDD_HHMMSS.bin`

Where:

- nodename is the DCFW node name;
- version is the UGOS version, and
- YYYYMMDD_HHMMSS is the settings export time in the UTC timezone, for example:

utm-utmcore@heashostatot_6.1.1.10462R-1_20210511_095942

In the **Device management → Settings export** section, click **Import**, and browse to the path of the settings file created earlier. The settings will be applied to the server, after which the server will reboot.

> **Note**
> **To correctly import the rules that use updatable UserGate lists (applications, URL categories, etc.), you need to have licenses for the SU and ATP modules as well as pre-downloaded UserGate lists.**

In addition, the administrator can configure a scheduled settings upload to external servers (FTP, SSH). To create a schedule for uploading settings, follow these steps:

1. Under **Device management → Settings export and import**, click **Add** and enter a name and description for the rule.

2. Specify the rule name and description in the **General** tab.

3. In the **Remote server** tab, specify the external server settings:

- **Server type**: FTP or SSH
- **Address**: the server's IP address
- **Port**: the server's port
- **Login name**: the user account on the remote server
- **Password/Repeat password**: the password for the user account
- **Directory path**: the path on the server where the settings will be uploaded The system itself will not create non-existent folders!

3. In the **Schedule** tab of the rule, specify when the settings should be uploaded. To set the time in the crontab format, specify it as follows: (minutes:0-59) (hours:0-23) (days of the month:1-31) (month:1-12) (day of the week:0-6, where 0 is Sunday).

Each of the first five fields can be defined using:

- An asterisk (*) denotes the entire range (from the first number to the last).
- A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7.
- Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23".
- The asterisk and dash are also used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

UserGate DCFW console access management

Access to the UserGate DCFW web console is controlled by creating additional administrator accounts, assigning them access profiles, defining an administrator password management policy, and configuring web console access with the service permissions in the network zone properties. As an additional security measure for console access, you can turn on certificate-based authorization for administrators.

> **Note**
>
> **A local superuser named Admin is created during the initial setup of DCFW.**

To create additional device administrator accounts, follow these steps:

Name	Description
Step 1. Create an administrator access profile.	In the **Administrators → Administrator profiles** section, click **Add** and enter the desired settings.
Step 2. Create an administrator account and assign it one of the administrator profiles created earlier.	In the **Administrators** section, click **Add** and select the desired option. • **Add local administrator**: create a local user, set a password for the user, and assign them one of the access profiles created earlier. • **Add LDAP user**: add a user from an existing domain. This requires a correctly configured LDAP connector in the **Authorization servers** section. When logging in to the administrative console, the username must be specified in the user@domain format. Assign this user a profile created earlier. • **Add LDAP group**: add a user group from an existing domain. This requires a correctly configured LDAP connector in the **Authorization servers** section. When logging in to the administrative console, the username must be specified in the user@domain format. Assign this user a profile created earlier. • **Add administrator with authorization profile**: create a user and assign them an administrator profile created earlier and an authorization profile (this requires correctly configured authorization servers).

When creating an administrator access profile, specify the following parameters:

Name	Description
Name	Profile name.
Description	Profile description.
API permissions	The list of objects available for access delegation when using the Application Programming Interfaces (API). The objects are described in the API documentation. The following access options are available: • No access • Read only • Read and write
Web console permissions	The list of web console tree objects available for delegation. The following access options are available: • No access • Read only • Read and write
CLI permissions	CLI access can be enabled here. The following access options are available: • No access • Read only • Read and write

A DCFW administrator can configure additional administrator account protection settings, such as password complexity and temporary account blocking upon exceeding the max number of authorization failures.

To configure the above settings, follow these steps:

Name	Description
Step 1. Configure the password policy.	In the **Administrators → Administrators** section, click **Configure**.
Step 2. Fill in the relevant fields.	Provide values for these fields: • **Strong password**: enables the additional password complexity settings presented below, such as Minimum length, Minimum uppercase letters, Minimum lowercase letters, Minimum digit letters, Minimum special characters, and Maximum characters repetition block.

Name	Description
	• **Number of invalid auth attempts**: the number of failed attempts to authenticate as an administrator after which the account is blocked for **Block time**. • **Block time**: the time for which the account is blocked.

Note

The advanced administrator account security settings apply only to local accounts. If an account from an external directory (such as LDAP) is selected as the device administrator, the security settings for that account are determined by that external directory.

The administrator can define the zones from which access to the web console service will be allowed (TCP port 8001).

Note

Web console access should not be allowed for zones connected to uncontrolled networks (e.g. the Internet).

To allow the web console service for a specific zone, go to the zone properties and allow access to the **Administrative console** service in the **Access control** section. For more details on configuring zone access control, see the section Zone Configuration.

As an additional security measure for console access, you can turn on certificate-based authorization for administrators.

To turn on this mode, follow these steps (this example uses the openssl utility):

Name	Description
Step 1. Create additional administrator accounts.	Configure the settings as described earlier in this chapter — for example, create an administrator account named Administrator54.
Step 2. Create or import an existing CA (Certification Authority) type certificate for web console authorization.	Create or import an existing CA certificate (the public key will suffice) as described in chapter Certificate Management. **Important!** The existing CA certificate is the certificate that was used directly to sign the administrator certificates and not the root certificate. For example, to create a CA using openssl, invoke these commands:

Name	Description
	`openssl req -x509 -subj '/C=RU/ST=Moscow/O=MyCompany /CN=ca.mycompany.com' -newkey rsa:2048 -keyout ca-key.pem -out ca.pem -nodes` `openssl rsa -in ca-key.pem -out ca-key.pem` The file ca-key.pem will store the private key for the certificate and the ca.pem file will store the public key. Import the public key into DCFW.
Step 3. Create certificates for the administrator accounts.	Create certificates for each of the administrators using 3rd party utilities, such as openssl. The **Common name** field in the certificate must match the administrator account name created in DCFW exactly. For openssl and the Administrator54 user, the commands will be as follows: `openssl req -subj '/C=SG/ST=Singapore/O=MyCompany /CN=Administrator54' -out admin.csr -newkey rsa:2048 -keyout admin-key.pem -nodes`
Step 4. Sign the administrator certificates created at Step 2 with the CA certificate.	Sign the administrator certificates with the web console CA certificate using 3rd party utilities, such as openssl. For openssl, the commands will be as follows: `openssl x509 -req -days 9999 -CA ca.pem -CAkey ca-key.pem -set_serial 1 -in admin.csr -out admin.pem` `openssl pkcs12 -export -in admin.pem -inkey admin-key.pem -out admin.p12 -name 'Administrator54 client certificate'` The admin.p12 file will contain the signed administrator certificate.
Step 5. Add the signed certificates to the OS from which the administrators will be authorized for web console access.	Add the signed administrator certificates (admin.p12 in our example) to the OS (or to the Firefox browser if it is used for console access) from which the administrators will be authorized for web console access. For more details, see the documentation for the relevant OS.

Name	Description
Step 6. Switch the web console to the X.509 certificate authorization mode.	In the **General settings** section, change the **Web console authentication mode** to **X.509 certificate**.

> **Note**
> **The web console authentication mode can be switched using CLI commands.**

The **Administrators → Administrator sessions** section displays all administrators who are logged in to the DCFW administrative web console. Any of the administrator sessions can be reset (closed) if necessary.

Clustering and High Availability

UserGate DCFW supports the following cluster types:

- **Configuration cluster**. Nodes combined into a configuration cluster support unified configuration within the cluster.
- **High Availability (HA) cluster**. **High Availability (HA) cluster**. Up to 4 configuration cluster nodes can be combined into a HA cluster that supports the Active-Active or Active-Passive operation modes. You can build several HA clusters.

Configuration cluster

A number of settings are specific to each cluster node, e.g., network interface configuration and IP addressing. Such parameters include:

- Log Analyzer,
- diagnostics,
- interface,
- gateways,
- DHCP,
- routes,

- OSPF,
- BGP.

To create a configuration cluster:

1. See the Initial Configuration chapter.

2. In the **Zones** section, create a new dedicated zone for cluster settings replication or use an existing one (**Cluster**). In the zone settings in the **Access control** subsection, enable the **Administrative console** and **Cluster** services.

Do not use zones whose interfaces are connected to untrusted networks (e.g., the Internet) for replication.

3. In the **Device Management** section of the **Cluster configuration** window, select the current cluster node and click the **Edit** button. Specify the IP address of an interface located in the zone you configured at Step 2.

4. In the **Configuration cluster** subsection, click **Generate secret code** and copy the generated code to the clipboard. The code is required for one-time authorization of a second node before adding it to the cluster.

5. Connect to the web console of the second cluster node and select the installation language. Specify the network interface that will be used to connect to the first cluster node and assign it an IP address. Both cluster nodes must be on the same subnet. Both cluster nodes must reside in the same subnet — e.g., as is the case when the eth2 interfaces of the two nodes are assigned IP addresses 192.168.100.5/24 and 192.168.100.6/24, respectively. Otherwise, you need to specify the IP address of the gateway through which the first cluster node will be accessible.

Specify the IP address of the first node configured at Step 3, enter the master node secret, and press the **Connect** button. If the IP addresses of the cluster configured at Step 2 are assigned correctly, the second node will be added to the cluster, and all the settings from the first cluster will be replicated on the second one.

The state of configuration cluster nodes can be determined from the color of the indicator next to the UserGate node name in the **UserGate → Device management → Configuration Cluster** section:

- Green: the node is online
- Yellow: the configuration cluster nodes are being synchronized
- Red: communication with this node is lost, the node is offline

6. In the web console for the second cluster node, go to the **Network → Interfaces** and assign a correct zone to each network interface. The zones and their settings are obtained as a result of data replication from the first cluster node.

7. Configure the gateways, routes, OSPF settings, and BGP settings specific to each cluster node.

> **Note**
> **When adding a node to the configuration cluster, the interface and gateway settings for connecting to the master node are explicitly specified. The IP address assignment type of this interface will be static.**

Up to four configuration cluster nodes can be combined into a HA cluster. There can be multiple HA clusters: for example, nodes A, B, C, and D within the configuration cluster can form two HA clusters, A-B and C-D.

A HA cluster can operate in two modes, **Active-Active** and **Active-Passive**. The state of cluster nodes can be determined from the color of the indicator next to the DCFW node name in the **General settings → UserGate → Device management → HA clusters** section:

- No colored indication: cluster node is available.
- **Step 1.** Perform initial configuration on the first cluster node.
- **Red**: no communication with the adjacent configuration nodes

Active-Passive HA Cluster

In the Active-Passive mode, one of the servers operates as the master node that processes traffic and the rest act as backup. On each of the cluster nodes, network interfaces are selected to which the administrator assigns virtual IP addresses. Transmitted between these interfaces are VRRP advertisements — messages that nodes use to exchange information about their state.

> **Note**
> **The Active-Passive mode supports user session synchronization, which provides user-transparent traffic switching between nodes, except for the sessions that use a proxy (e.g., HTTP/S).**

When a backup server assumes the master role, **all** virtual IP address of **all** cluster interfaces are transferred to it. An unconditional role transfer occurs under the following circumstances:

- A backup server gets no confirmation that the master node is online (Timers: Adver - 1sec, MasterDown - 3sec) — for example, if it is offline or the nodes are unavailable on the network.
- Internet connectivity checking is configured on the node (see section Gateway Configuration), and there is no Internet access through any of the gateways.

  If the node specified in the network checker properties is unavailable at all cluster nodes, the HA cluster will be brought offline.
- A software fault has occurred in UserGate.

When one or more network interfaces that are assigned virtual IP addresses go offline, this will lower the node's priority but not necessarily cause a change in the server's role. Transition to a backup node will occur if that node has a higher priority than the master node. By default, the master node has a priority of 250, while a backup node has a priority of 249. A node's priority is decreased by 2 for each cluster interface that has no physical connectivity to the network. A node's priority is decreased by 2 for each cluster interface that has no physical connectivity to the network. Therefore, for a two-node HA cluster, if one network interface on the master node loses the physical connectivity to the network, the master role will be transferred to the backup server, provided that all its cluster interfaces have network connectivity (the priority value will be 248 for the master and 249 for the backup in that case).

If one or more cluster network interfaces go offline **on a backup node**, the node's priority will be lowered, but it will nevertheless be able to become the master in case of an unconditional role transfer or when the master node's priority drops below the priority of this backup node.

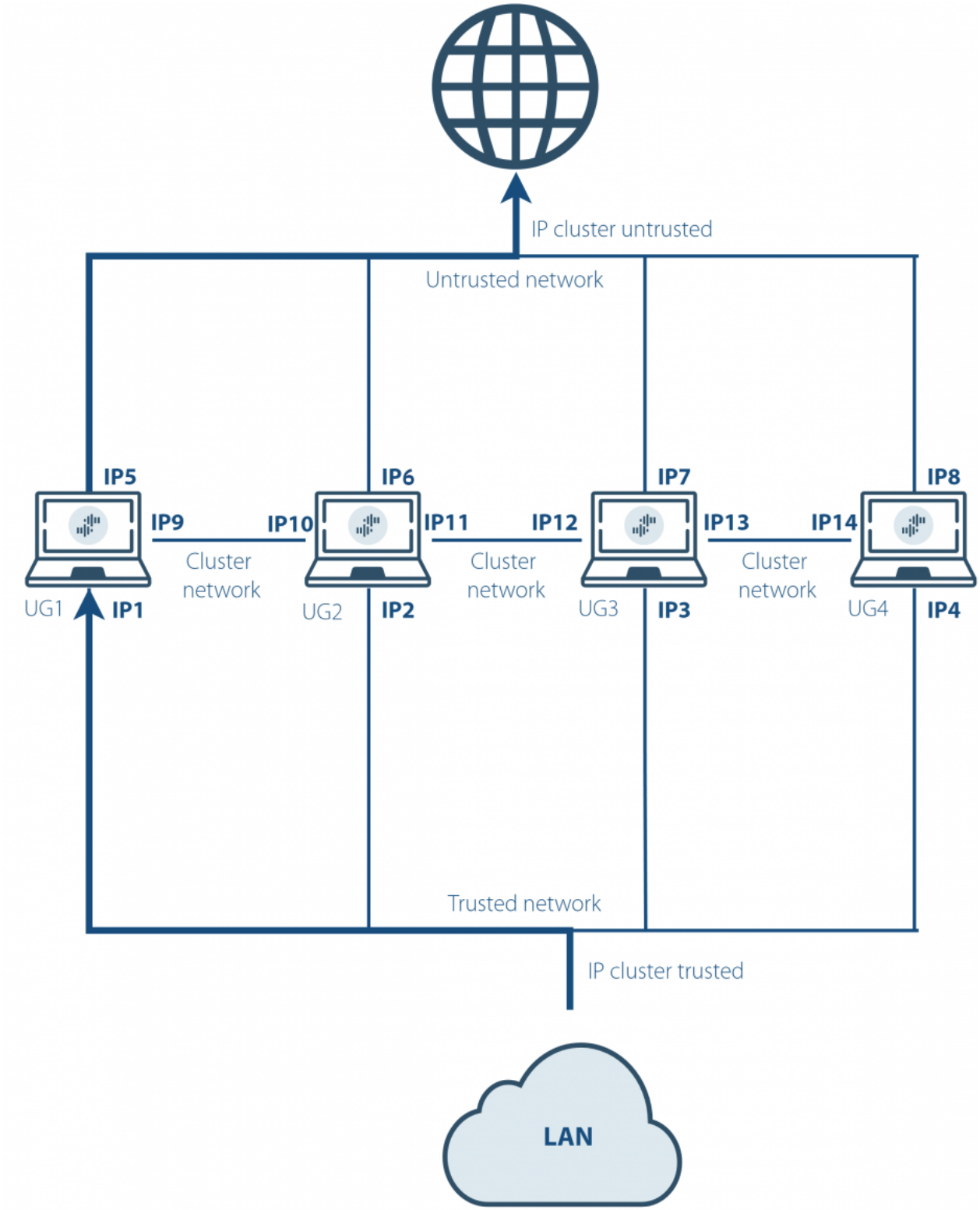
> **Note**
>
> **If cluster IP addresses are assigned to VLAN interfaces, the lack of connectivity on a physical interface will be interpreted by the HA cluster as a connectivity loss on all VLAN interfaces created on that physical interface.**

> **Note**
> **To reduce the time it takes for the network equipment to switch the traffic to a backup node, DCFW sends an internal GARP notification (Gratuitous ARP) to inform the network equipment of a MAC address change for all virtual IP addresses. DCFW sends a GARP packet every minute and when the master role is transferred to a backup server.**

An example network diagram for a HA cluster in the Active-Passive mode is shown below. The network interfaces are configured as follows:

- **Trusted zone**: IP1, IP2, IP3, IP4, and IP cluster (Trusted).
- **Untrusted zone**: IP5, IP6, IP7, IP8, and IP cluster (Untrusted).
- **Cluster zone**: IP9, IP10, IP11, IP12, IP13, IP14. The interfaces in the Cluster zone are used for parameter replication.

Both cluster IP addresses reside on the UG1 node. Both cluster IP addresses reside on the UG1 node.



Active-Active HA Cluster

In the Active-Active mode, one of the servers operates as the master node that distributes the traffic among all other cluster nodes. On each of the cluster nodes, network interfaces are selected to which the administrator assigns virtual IP

addresses. Transmitted between these interfaces are VRRP advertisements — messages that nodes use to exchange information about their state.

Virtual IP addresses always reside on master node interfaces, therefore the master node receives and responds to client ARP requests, consecutively serving MAC addresses of all nodes of the HA cluster to ensure uniform traffic distribution to all cluster nodes with consideration of the need to provide user session continuity.

> **Note**
> **The Active-Active mode supports user session synchronization, which provides user-transparent traffic switching between nodes, except for the sessions that use a proxy (e.g., HTTP/S).**

When a backup server assumes the master role, **all** virtual IP address of **all** cluster interfaces are transferred to it. An unconditional role transfer occurs under the following circumstances:

- A backup server gets no confirmation that the master node is online (Timers: Adver - 1sec, MasterDown - 3sec) — for example, if it is offline or the nodes are unavailable on the network.
- Internet connectivity checking is configured on the node (see section Gateway Configuration), and there is no Internet access through any of the gateways.
- A software fault in DCFW.

When one or more network interfaces on the **master node** that are assigned virtual IP addresses go offline, this will lower the node's priority but not necessarily cause a change in the server's role. Transition to a backup node will occur if that node has a higher priority than the master node. By default, the master node has a priority of 250, while a backup node has a priority of 249. A node's priority is decreased by 2 for each cluster interface that has no physical connectivity to the network. Therefore, for a two-node HA cluster, if one network interface on the master node loses the physical connectivity to the network, the master role will be transferred to the backup server, provided that all its cluster interfaces have network connectivity (the priority value will be 248 for the master and 249 for the backup in that case). When the physical connectivity on the original master node is restored, that node will assume the master role again because its priority value will return to 250.

When one or more cluster network interfaces go offline **on a backup node**, this lowers the node's priority and excludes it from traffic load balancing. That backup node will nevertheless be able to become the master in case of an unconditional

role transfer or when the master node's priority drops below the priority of this backup node.

> **Note**
> **If cluster IP addresses are assigned to VLAN interfaces, the lack of connectivity on a physical interface will be interpreted by the HA cluster as a connectivity loss on all VLAN interfaces created on that physical interface.**

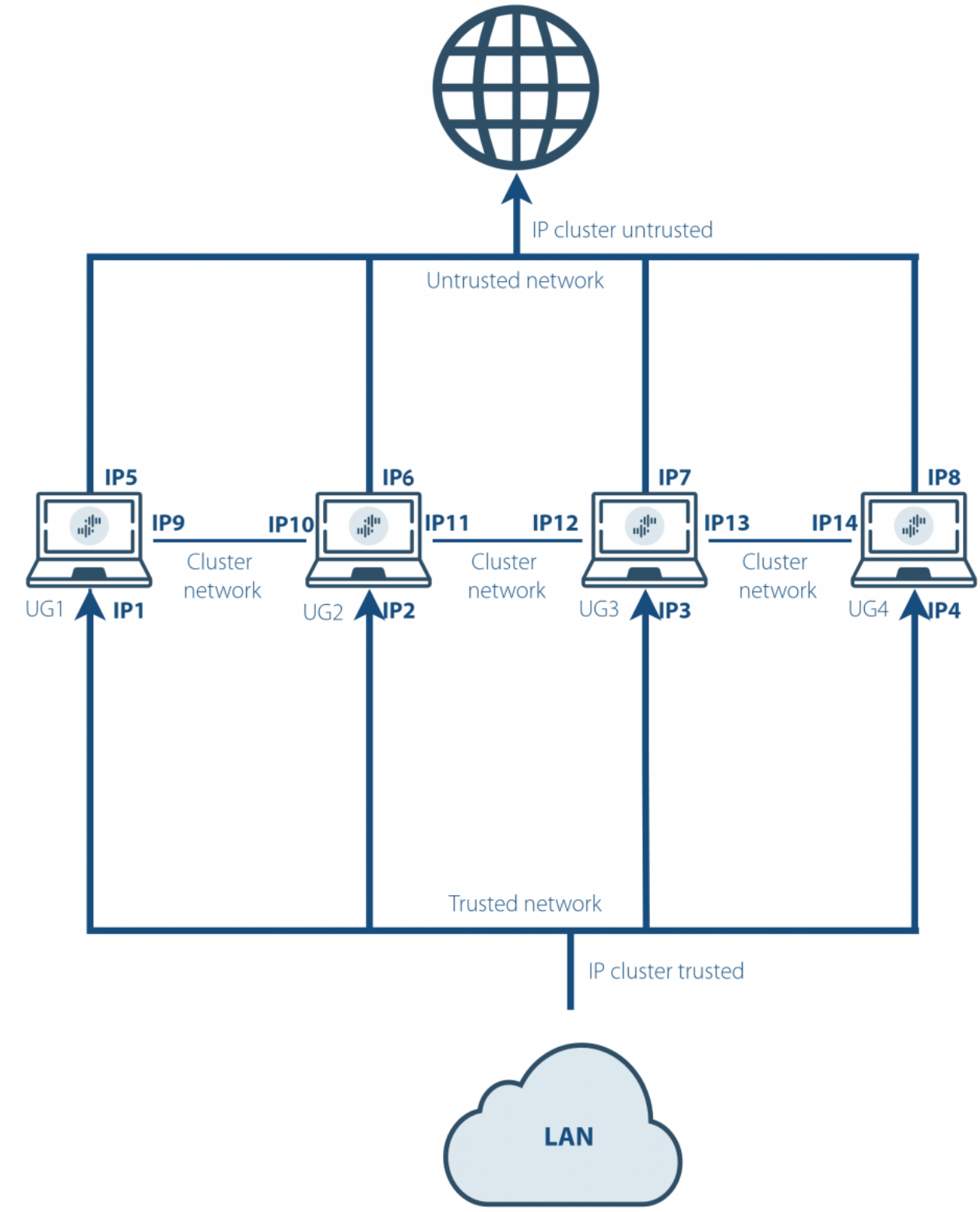
Note

To reduce the time it takes for the network equipment to switch the traffic to a backup node, DCFW sends an internal GARP notification (Gratuitous ARP) to inform the network equipment of a MAC address change for all virtual IP addresses. In the Active-Active mode, DCFW sends a GARP packet only when a backup server assumes the master role.

An example network diagram for a HA cluster in the Active-Active mode is shown below. The network interfaces are configured as follows:

- **Trusted zone**: IP1, IP2, IP3, IP4, and IP cluster (Trusted).
- **Untrusted zone**: IP5, IP6, IP7, IP8, and IP cluster (Untrusted).
- **Cluster zone**: IP9, IP10, IP11, IP12, IP13, IP14. The interfaces in the Cluster zone are used for settings replication.

Both cluster IP addresses reside on the UG1 node. If the UG1 node goes offline, both cluster IP addresses will migrate to the next server, which becomes the master — e.g., UG2.



A HA cluster in the Active-Active mode

> **Note**
> **For correct traffic processing, the reverse traffic from the server to the client must pass through the same DCFW node that was used for the corresponding forward traffic from the client, i.e., the user session must always pass through the same cluster node. The simplest solution is to use NAT from the client network to the server network (NAT from Trusted to Untrusted).**

To create a HA cluster, follow these steps:

Name	Description
Step 1. Create a configuration cluster.	Create a configuration cluster as described in the previous step.
Step 2. Configure zones whose interfaces will participate in the HA cluster.	In the **Zones** section, you should allow the VRRP service for all zones where virtual cluster IP addresses are to be added (zones **Trusted** and **Untrusted** on the above diagrams).
Step 3. Create a HA cluster.	In the **Device management → HA cluster** section, click **Add** and configure the settings for the new HA cluster.
Step 4. Specify a virtual IP address for the auth.captive, logout.captive, block.captive, and ftpclient.captive hosts.	If captive-portal authorization is to be used, the system host names auth.captive and logout.captive used by the authorization procedures in the captive portal must resolve to the IP address assigned as the virtual cluster address. For more details on these settings, see the section General Settings.

The settings for a HA cluster are listed below:

Name	Description
Enabled	Enable or disable the HA cluster.
Name	The name of the HA cluster.
Description	A description of the HA cluster.
Mode	The HA cluster operating mode: • **Active-Active**: the load is distributed between all cluster nodes. • **Active-Passive**: the load is processed by the master node and switched to a backup instance if the master node is offline.

Name	Description
Sessions sync	Enables user session synchronization mode between all nodes in the HA cluster. When enabled, this option makes switching users between devices transparent to the users themselves but adds significant load on the UserGate platform. The option is only relevant for the Active-Passive cluster mode.
HA cluster multicast ID	Multiple HA clusters can be created in a single configuration cluster. Session synchronization uses a specific multicast address defined by this parameter. A unique ID must be assigned to each group of HA clusters that requires session synchronization support within the group.
Virtual router ID (VRID)	The VRID must be unique to each VRRP cluster in the local network. If there are no 3rd party VRRP clusters in the network, it is recommended to keep the default setting.
Nodes	Select the configuration cluster nodes to combine into an HA cluster. Here you can also assign the master role to one of the selected nodes.
Virtual IPs	Assign virtual IP addresses and map them to the interfaces of the cluster nodes.
UPD/ICMP Synchronization	Manage the user session synchronization mode: • **Synchronize all sessions**: enable/disable synchronizing all user sessions, including UDP/ICMP sessions. If this is disabled and the **Sessions sync** setting on the **General** tab is enabled, only TCP sessions will be synchronized. • **IPs excluded from synchronization**: list the IP addresses for which user sessions will not be synchronized.

Certificate Management

General Information

UserGate DCFW uses the secure HTTPS protocol for device management, and it can also authorize administrators in the web console using their certificates.

To perform these functions, DCFW employs different types of certificates:

Name	Description
Web console SSL certificate	Used to create a secure HTTPS administrator connection to a DCFW web console.
Captive portal SSL certificate	Used to create a secure HTTPS user connection to the captive portal auth page, to display a block page or the captive portal's logout page, and to support FTP proxy operation. This certificate must be issued with the following parameters: • Subject name — the value set for the **Captive portal auth domain** defined on the **General settings** page. • Subject Alternative names — include all domains for which this certificate is used as they are specified on the **General settings** page: ◦ Captive portal Auth domain ◦ Captive portal Logout domain ◦ Block page domain ◦ FTP over HTTP domain ◦ Web portal domain specified in the web portal settings. By default, a certificate for the auth.captive domain signed with an SSL inspection certificate is used with the following parameters: • Subject name = auth.captive • Subject alternative names = auth.captive, logout.captive, block.captive, ftpclient.captive, sslvpn.captive If the administrator has not loaded their own certificate for this role, DCFW will automatically reissue this certificate when the administrator changes one of the domains on the **Settings** page (those used for auth.captive, logout.captive, block.captive, ftpclient.captive, and sslvpn.captive). **Note** **If the administrator uses a separate certificate for the Captive portal domain, then he must add not only his Auth captive portal domain, but also the fixed cert.captive domain in the Subject Alternative name extension of the certificate. If cert.captive is not added, the browser will generate a security error when authenticating via a certificate.**

Name	Description
User certificate	The certificate assigned to a DCFW user. The user can be either added locally or imported from LDAP. The certificate can be used to authorize user access to the published resources using reverse proxy rules.
Web console certification chain	Certificate authority certificate for access to the web console. For successful authorization, the administrator certificate must be signed with a certificate of this type.
SAML server	Used to enable DCFW operation with a SSO SAML IDP server. For more details on configuring the DCFW to work with a SAML IDP authorization server, see the relevant section of this Guide.

There can be multiple web console SSL and captive portal SSL certificates, but only one certificate of each type may be active and used for respective purposes. There can also be multiple **web console authorization CA** type certificates, and each of them can be used to verify the authenticity of administrator certificates.

To create a new certificate, follow these steps:

Name	Description
Step 1. Create a new certificate.	In the **Certificates** section, click **Create**.
Step 2. Fill in the relevant fields.	Provide values for these fields: • **Name**: the name under which the certificate will be displayed in the certificate list. • **Description**: a description of the certificate. • **Country**: the country where the certificate is being issued. • **State or province name**: the state or province where the certificate is being issued. • **Locality name**: the city or town where the certificate is being issued. • **Organization name**: the name of the organization to which the certificate is being issued. • **Common name**: the certificate name. To ensure compatibility with the majority of browsers, we recommend using only Latin characters. • **Email**: your company's email.
Step 3. Specify the purpose of the certificate.	After creating the certificate, specify its intended role in DCFW. To do this, select the relevant certificate in the certificate list,

Name	Description
	click **Edit**, and specify the type of the certificate (web console SSL certificate, web console authorization CA). If you selected a web console SSL certificate, DCFW will reboot the web console service and prompt you to connect using the new certificate.

DCFW allows you to export certificates created there and import certificates created in other systems — e.g., a certificate issued by a CA that your organization trusts.

To export a certificate, follow these steps:

Name	Description
Step 1. Select a certificate for export.	Select the desired certificate in the certificate list.
Step 2. Export the certificate.	Select the export type: • **Export certificate**: export certificate data in the .der format without exporting the certificate's private key. Use the exported SSL inspection certificate file to set it as the local CA on user computers. For more details on this, see the Installing local CA certificates appendix. • **Export CSR**: export a CSR, e.g., to be signed by a CA.

Note

It is recommended to save the certificate to be able to restore it later.

Note

For security purposes, UserGate does not allow the export of private keys for certificates.

Note

Users can download an SSL inspection certificate for installation on their own computers from the UserGate server from a direct link: http://UserGate_IP:8002/cps/ca

To import a certificate, you need to have the certificate files (and, optionally, the private key for the certificate). If you have those, follow the steps below:

Name	Description
Step 1. Start the import procedure.	Click **Import**.
Step 2. Fill in the relevant fields.	Provide values for these fields: • **Name**: the name under which the certificate will be displayed in the certificate list. • **Description**: a description of the certificate. • **Certificate file**: upload the certificate data file. • **Private key**: upload the private key file for the certificate. • **Passphrase**: specify the private key passphrase (if required). • **Certificate's chain**: a file containing the upstream CA certificates used when creating this certificate. This field is optional.

Client Certificate Profiles

A client certificate profile allows managing certificates that provide the security and authentication of network connections. The profile specifies CA certificates, methods for checking the relevance of user certificates, and methods for selecting a user name for authentication.

The client certificate profile is used to validate the certificate provided by the client. The client certificate is checked for validity for each CA certificate in the list.

When certificate-based (PKI) authentication is selected, a preconfigured client certificate profile is specified pointing to certificates that can then be used in various DCFW subsystems, such as Captive portal, VPN, web portal, and reverse proxy.

To create a client certificate profile, go to **Settings → UserGate → Client certificate profiles**, click **Add**, and specify the desired settings:

Name	Description
Name	The name of the client certificate profile.
Description	Optional profile description.

Name	Description
Get username from	Select the field in the certificate that determines the username used for authentication: • Common-name: the domain name or hostname in the Subject field for which the certificate is intended. • Subject altname email: the parameter with the email prefix in the SAN (Subject Alternative Name) extension is used to determine the username. • Principal name: the Universal Principal Name (UPN) parameter contained in the otherName field in the SAN extension is used to determine the username. If multiple UPNs or email addresses are specified in the SAN extension fields of a certificate, the first one specified in the certificate is used.
CA certificates	The CA certificates assigned to the profile. List of Certification Authority certificates. Used to validate the client-supplied certificate. The client certificate is checked for validity for each CA certificate in the list. The list is iterated from top to bottom.
Checking revoked certificates	Certificate revocation lists (CRLs) contain certificates that have been revoked and can no longer be used. This list includes certificates that have expired or been compromised. Certificate revocation status check method: • Do not check: do not check any certificate. • The whole chain: check all certificates in the chain and require that they are all valid. • User certificate: check only the client certificate. • Consider valid if the status is unknown: if the CRL could not be verified for some reason, then the certificate is considered valid (however, it is still checked and may return the invalid status if the certificate is on the revocation list).
Check timeout	The time interval after which DCFW stops waiting for the response from the certificate revocation list service.

Expanding the System Partition

To expand the system partition while preserving the configuration and data of the UserGate node, follow these steps:

Name	Description
Step 1. Add an new virtual disk.	Use the hypervisor to add a **new** disk of the required size in the UserGate virtual machine properties.
Step 2. Expand the partition size in the system utilities.	In the UserGate node boot menu, enter the **Support menu** section. In the section that opens, select **Expand data partition** and start the partition expansion process.
Step 3. Check the size of the system partition.	When the expansion process is complete, boot the node and check the size of the system partition in the **Dashboard → Disk s** section.

Note

Expanding the system partition by increasing the size of the *existing* virtual machine disk is only possible if you reset the node to factory settings, i.e. perform a factory reset.

System utilities

Administrators can access system utilities in the boot menu during the DCFW booting process. To access this menu, connect a display to a VGA or HDMI port, a keyboard to a USB port (if a device has these ports), or connect your computer to DCFW using a serial port cable or a USB-Serial adapter. Launch a terminal that supports connecting via a serial port, e.g. Putty for Windows. Establish a serial port connection using 115200 8n1 as the connection parameters.

During the boot process, the administrator can select from the following boot menu options:

Name	Description
UGOS DCFW	Boot DCFW and output the boot process diagnostic information to the serial port.

Name	Description
UGOS DCFW (failsafe)	Boot DCFW in a simplified video mode.
Support menu	Enter the system utilities section and send output to tty1 (the monitor).
Restore previous version	This section is available after updating or creating a system backup.

The system utilities (Support menu) section offers the following actions:

Name	Description
Check filesystems	Start a file system check on the device with automatic error correction.
Expand data partition	Expand the data partition. This operation is usually carried out after increasing the amount of disk space allocated by the hypervisor to the DCFW VM. **Important!** To expand the system partition while preserving DCFW data and settings, you need to add a **new** disk by using the hypervisor, and then perform the **Expand data partition** operation, as described in the Expanding the system partition article of the administrator guide.
Create backup	Create a full backup of the DCFW disk on an external USB drive. **Important!** Your USB drive will be formatted before creating a backup.
Restore from backup	Restore DCFW from an external USB drive.
Factory reset	Reset the DCFW status. The software version will remain the same as the one installed when the command was run. All data and settings will be lost.
Exit	Log out and reboot the device.

NETWORK CONFIGURATION

Zone Configuration

A zone in DCFW is a logical aggregation of network interfaces. DCFW security policies use interface zones instead of interfaces themselves. This provides the needed flexibility to the policies and significantly eases the management of a HA cluster. Zones are the same on all cluster nodes, i.e., this is a global setting for the entire cluster.

It is recommended to aggregate interfaces into a zone based on their intended use, e.g., a LAN interface zone, Internet interface zone, partner-connected interface zone, etc.

DCFW is supplied with the following default zones:

Name	Description
Management	Used to connect trusted networks from which DCFW management is allowed.
Trusted	Used to connect trusted networks, such as LANs.
Untrusted	Used for interfaces connected to untrusted networks, such as the Internet.
DMZ	Used for interfaces connected to the DMZ network.
Cluster	Used for interfaces that support the operation of a cluster.
VPN for Site-to-Site	Used for all Office-to-Office clients that connect to DCFW using a VPN.
VPN for remote access	Used for all mobile users who connect to DCFW using a VPN.
Tunnel inspection zone	Tunnel inspection zone. All source and destination addresses of packets encapsulated into a tunnel will belong to this zone.

DCFW administrators can edit the settings for the default zones and create additional zones.

> **Note**
> **A maximum of 255 zones can be created.**

To create a zone, follow these steps:

Name	Description
Step 1. Create a new zone.	Click **Add** and provide a name for the new zone
Step 2. (Optional) Configure the DoS protection settings for the zone.	Configure the network flood protection settings for TCP (SYN-flood), UDP, and ICMP protocols in the zone: • **Aggregate**: if set, all incoming packets to the zone's interfaces are included in the count. If not set, packets are counted separately for each IP address. • **Alert threshold**: when the number of requests exceeds this threshold, the event is recorded in the system log. • **Drop threshold**: when the number of requests exceeds this threshold, DCFW starts dropping the packets and records the event in the system log. The recommended values are 3000 requests per second for the alert threshold and 6000 requests per second for the drop threshold. It is recommended to enable flood protection on all interfaces except those in the **Cluster** zone. The UDP drop threshold should be increased if the zone's interfaces carry traffic for services such as VoIP or L2TP VPN. **DoS protection exclusions**: here you can list the server IP addresses that need to be excluded from the protection. This can be useful, e.g., for the VoIP service as it sends large numbers of UDP packets.
Step 3. (Optional) Configure the access control settings for the zone.	Specify the DCFW-provided services that will be available to clients connected to this zone. It is recommended to disable all services for zones connected to uncontrolled networks, such as the Internet. The following services exist: • **Ping**: enables pinging of DCFW. • **SNMP**: provides SNMP access to DCFW (UDP 161). • **Captive portal** and **Block pages**: required for displaying the captive portal's auth page and block page (TCP 80, 443, 8002). • **API XML RPC over HTTP**: allows you to control the device via API (TCP 4040). • **Cluster**: required for combining several DCFW nodes into a cluster (TCP 4369, TCP 9000-9100). • **VRRP**: required for combining several DCFW nodes into a HA cluster (IP protocol 112). • **Administrative console**: provides access to the administrative web console (TCP 8001). • **DNS**: provides access to the DNS proxy service (TCP 53, UDP 53).

Name	Description
	• **HTTP(S) proxy**: provides access to the HTTP(S) proxy service (TCP 8090). • **Authorization agent**: provides server access required by Windows authorization agents and terminal servers (UDP 1813). • **CLI over SSH**: provides server access for management using CLI (command line interface) (TCP port 2200). • **VPN**: provides server access for connecting L2TP VPN clients (UDP 500, 4500). • **Log Analyzer/SIEM**: provides connection to Log Analyzer (TCP 2023, 9713). • **OSPF**: OSPF dynamic routing service. For more details, see the OSPF section. • **BGP**: BGP dynamic routing service. For more details, see the BGP section. • **RIP**: RIP dynamic routing service. • **BFD**: quick network connection failure detection service. • **SNMP Proxy**: service used to build a distributed monitoring system for load balancing and distributed network infrastructure monitoring. • **Multicast**: multicast service. • **NTP service**: enables access to a time service running on the DCFW server. • **UserID syslog collector**: a service that enables information collection from remote devices using the Syslog protocol (the default port number is 514). • **Endpoints connection**: a service used to allow connection of endpoints with UserGate Client software (TCP 4045) installed. For more on network availability requirements, see the appendix Network Environment Requirements.
Step 4. (Optional) Configure the IP spoofing protection settings.	IP spoofing attacks allow a malicious actor to transmit a packet from an external network, such as **Untrusted**, to an internal one, such as **Trusted**. To do that, the attacker substitutes the source IP address with an assumed internal network address. In this case, responses to this packet will be sent to the internal address. To protect against this kind of attack, the administrator can specify the source IP address ranges allowed in the selected zone. Network packets with source IP addresses other than those specified will be discarded. Using the **Negate** checkbox, the administrator can specify the source IP addresses from which packets may not be received on this zone's interfaces. In this case, packets with source IP

Name	Description
	addresses within those ranges will be rejected. As an example, for the Untrusted zone, you can specify "gray" IP address ranges as 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 and turn on the **Negate** option.
Step 5. (Optional) Set session limits.	Limiting the number of concurrent connections from a single IP address is a security measure that limits active network connections originating from the same IP. This is done for several reasons: • To defend from attacks: malicious users can use a large number of concurrent connections from one IP address to launch DDoS attacks (distributed attacks that aim to cause denial of service). Limiting the number of such connections helps lower the risks of these attacks by reducing the network or server load. • To prevent abuse: some users may try to abuse the resources by creating many concurrent connections. Limiting connections helps prevent resource overuse and maintain a uniform load distribution. • To preserve availability: preventing situations when one user takes up all available resources, leaving little for others. The limits help preserve resource availability for all users. • To better manage resources: more efficient network and server resource management ensures a more stable and predictable performance. To limit the number of concurrent connections from a single IP address: 1. Set the **Enable sessions limiting per IP** checkbox. 2. Specify the maximum allowed number of sessions originating from a single IP address. 3. Add a list of IP addresses to which the limit will not apply. For more details about how to create an IP address list, see the IP Addresses section.

Network Interface Configuration

The **Interfaces** section displays all physical and virtual network interfaces existing in the system and allows you to modify their settings and add VLAN interfaces. All interfaces of each cluster node are displayed here. The interface settings are node-specific — that is, they are not global.

Using the **Edit** button, you can modify the settings for a network interface:

- Enable or disable the interface
- Specify the interface type as Layer 3 or Mirror. An interface operating in the Layer 3 mode can be assigned an IP address and used in firewall rules, content filtering, and other rules. This is the standard operating mode of a network interface. An interface operating in the Mirror mode can receive traffic from a SPAN port of network equipment for subsequent analysis.
- Assign a zone to the interface
- Assign a Netflow profile to send statistics to a Netflow collector.
- Assign a profile for sending data using the Link Layer Discovery Protocol (LLDP). Available only for adapter type interfaces.
- Assign an alias, which is an additional identifier for an interface. This optional setting is used for working with SNMP.
- Modify the physical parameters of the interface, such as the MAC address and MTU size.
- Select the IP address assignment type: no address, a static IP address, or a dynamic IP address obtained using DHCP.
- Configure DHCP relay for the selected interface. To do this, you need to enable DHCP relay, enter the IP address of the interface on which the relay is added in the **UserGate address** field, and specify one or more DHCP servers where client DHCP requests are to be forwarded.

Using the **Add** button, you can add the following logical interface types:

- VLAN
- Bond.
- Bridge
- PPPoE
- VPN
- Tunnel.
- Loopback

Creating a VLAN Interface

Using the **Add VLAN** button, the administrator can create sub-interfaces. To create a VLAN, provide the following settings:

Name	Description
Enabled	Enables the VLAN.
Name	The VLAN name. Assigned automatically based on the physical port name and the VLAN tag.
Description	An optional interface description.
Type	Specify the interface type as Layer 3 or Mirror. An interface operating in the Layer 3 mode can be assigned an IP address and used in firewall rules, content filtering, and other rules. This is the standard operating mode of a network interface. An interface operating in the Mirror mode can receive traffic from a SPAN port of network equipment for subsequent analysis.
VLAN tag	The sub-interface number. Up to 4094 interfaces can be created.
Node name	The node name in the cluster where this VLAN is being created.
Interface	The physical interface on which the VLAN is being created.
Zone	The zone to which the VLAN belongs.
Netflow profile	The Netflow profile to send statistical data to the Netflow collector. You can read about Netflow profiles in the Netflow Profiles chapter.
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
Networking	The IP address assignment method: no address, a static IP address, or a dynamic IP address obtained using DHCP. The capability to change a MAC address, MTU size, and MSS size.
DHCP relay	Configure DHCP relay for a VLAN interface. Enable DHCP relay, enter the IP address of the interface on which the relay function is added in the **UserGate address** field, and specify one or more DHCP servers where client DHCP requests are to be forwarded.

Bonding Network Interfaces

Using the **Add bond** button, the administrator can bond several physical network interfaces into a single aggregated logical interface to increase the bandwidth or provide high availability. To create a bond, provide the following settings:

Name	Description
Enabled	Enables the bond.
Name	The bond name.
Node name	The DCFW cluster node on which the bond will be created.
Zone	The zone to which the bond belongs.
Netflow profile	The Netflow profile to send statistical data to the Netflow collector. You can read about Netflow profiles in the Netflow Profiles chapter.
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
Interfaces	One or more network interfaces that will be used to create the bond.
Aggregation mode	The aggregation mode must match the operating mode for the device to which the bond is connected. The options are: • **Round robin.** Packets are sent consecutively, starting from the first available slave and continuing to the last one. This policy is used to provide load balancing and high availability. • **Active backup**. Only one network interface in the bond will be active. Another slave interface can become active only if the currently active interface fails. With this policy, the MAC address of the bond interface is only visible externally through one network port to avoid problems with the switch. This policy is used for high availability. • **XOR.** Transmission is distributed between the slave interfaces using the formula: [(XOR) MOD]. This means that the same NIC sends packets to the same recipients. Optionally, the transmission allocation can also be based on the xmit_hash policy. The XOR policy is used to provide load balancing and high availability. • **Broadcast.** Transmits everything on all network interfaces. This policy is used for high availability.

Name	Description
	• **IEEE 802.3ad**. The default mode, supported by most network switches. Creates aggregated groups of NICs with identical speed and duplex settings. When combined like this, all links in the active aggregation participate in transmission as per IEEE 802.3ad. The choice of interface for packet transmission is determined by the policy. By default, the XOR policy is used, with the xmit_hash policy as a possible alternative. • **Adaptive transmit load balancing.** The outgoing traffic is distributed depending on the load on each slave interface (determined by the download speed). No additional configuration on the switch is required. The incoming traffic is received by the current network card. If this card fails, another card assumes the MAC address of the failed one. • **Adaptive load balancing.** Includes the previous policy plus incoming traffic balancing. No additional configuration on the switch is required. The incoming traffic is balanced through ARP negotiation. The driver intercepts ARP responses sent from the local NICs to the outside and overwrites the source MAC address with one of the unique MAC addresses of the NIC in the bond. Thus, different peers use different server MAC addresses. The incoming traffic is balanced sequentially (round-robin) among the interfaces.
MII monitoring period (msec)	Sets the MII monitoring period in milliseconds. Determines how often the link state will be checked for failures. The default value of 0 disables MII monitoring.
Down delay (msec)	Sets the delay in milliseconds before disabling the interface on a connection failure. This option is only valid for MII monitoring (miimon). The parameter value must be a multiple of miimon, otherwise it will be rounded to the nearest multiple. Default value: 0.
Up delay (msec)	Sets the delay in milliseconds before bringing up the link on discovering that it has been restored. This parameter is only valid with MII monitoring (miimon). The parameter value must be a multiple of miimon, otherwise it will be rounded to the nearest multiple. Default value: 0.

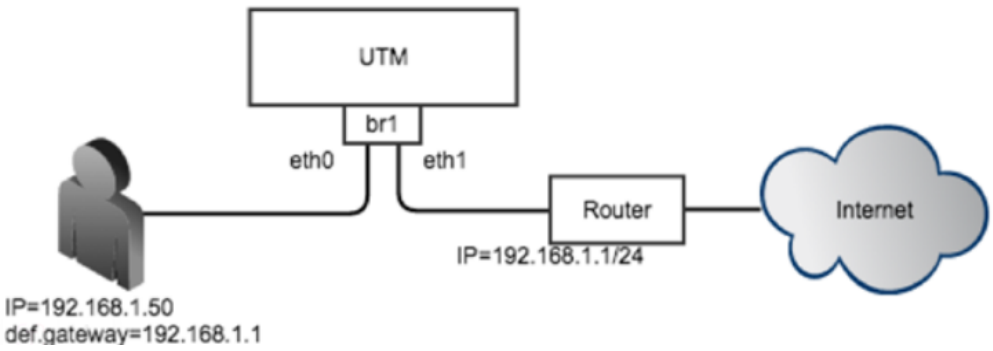
Name	Description
LACP rate	Determines the interval between LACPDU packets sent by the partner in the 802.3ad mode. Enumerated options: • **Slow**: requests that the partner send LACPDU packets every 30 seconds. • **Fast**: requests that the partner send LACPDU packets every second.
Failover MAC	Determines how MAC addresses will be assigned to the bonded slaves in the active-backup mode on switching between slaves. The normal behavior is to use the same MAC address on all slaves. Enumerated options: • **Disabled**: sets the identical MAC address on all slaves during the switching process. • **Active**: the MAC address on the bond interface will always be identical to that on the currently active slave. The MAC addresses on the backup interfaces are not changed. The MAC address on the bond interface changes during the failover processing. • **Follow**: the MAC address on the bond interface will be the same as that on the first slave added to the bond. This MAC is not set on the second and subsequent interfaces while they are in backup mode. That MAC address gets assigned during a failover: when a backup slave interface becomes active, it assumes a new MAC (the one on the bond interface), and the formerly active slave is assigned the MAC that the currently active one used to have.
Xmit hash policy	Determines the hash policy for packet transmission via bonded interfaces in the XOR or IEEE 802.3ad modes. Enumerated options: • **Layer 2**: only MAC addresses are used for hash generation. With this algorithm, the traffic for a particular network host is always sent over the same interface. This algorithm is compatible with IEEE 802.3ad. • **Layer 2+3**: both MAC and IP addresses are used for hash generation. This algorithm is compatible with IEEE 802.3ad. • **Layer 3+4**: IP addresses and transport-layer protocols (TCP or UDP) are used for hash generation. This algorithm is not universally compatible with IEEE 802.3ad, as both fragmented and non-fragmented packets can be transmitted within a single TCP or UDP interaction. Fragmented packets lack the source and destination ports. As a result, packets from the same session can

Name	Description
	reach the recipient in an order other than the intended one because they are sent via different slaves.
Networking	The IP address assignment method: no address, a static IP address, or a dynamic IP address obtained using DHCP. The capability to change a MAC address, MTU size, and MSS size.
DHCP relay	This is used to configure DHCP relay for the bond interface. Enable DHCP relay, enter the IP address of the interface on which the relay function is added in the **UserGate address** field, and specify one or more DHCP servers where client DHCP requests are to be forwarded.

Interface Bridging

A network bridge works at the link layer (L2) of the OSI networking model. When the bridge receives a network frame, it checks the frame's MAC address and, if the MAC does not belong to the same subnet, passes (forwards) this frame further; if the frame belongs to the same subnet, the bridge does nothing.

A bridge interface can be used in DCFW like a regular network interface. Moreover, you can use a bridge to configure in-transit content filtering at L2 without introducing any changes to the corporate IT infrastructure. The simplest schema for using DCFW as an L2 content filtering solution looks like this:



When creating a bridge, you can specify the operating mode for it as Layer 2 or Layer 3.

If Layer 2 is selected, the bridge does not need to be assigned an IP address, routes, or gateways for it to work correctly. In this mode, the bridge works at the MAC address level by forwarding packets from one network segment to another. SCADA and Mail security rules cannot be used in this scenario, but content filtering works.

Important!

The DNS filtering and L2 bridge functionality are not compatible in the current version: when DNS filtering is enabled, DNS requests stop passing through the bridge.

If Layer 3 is selected, you need to assign the bridge an IP address and specify routes in networks connected to the bridge's interfaces. In this mode, all filtering mechanisms available in DCFW can be used.

If the bridge is created in a DCFW HSC equipped with a network card that supports the bypass mode, you can combine two interfaces into a bypass bridge. The bypass bridge automatically switches two selected interfaces to the bypass mode (bridging them so that all traffic bypasses DCFW) if:

- The DCFW HSC is powered off.
- The self-diagnostics system has encountered a runtime problem in DCFW software. The timeout for detecting a problem is 10 seconds.

Control of the bypass relay operation of network ports is possible via the PMC interface. For more information, see the Platform Management Commands section of the PMC CLI Guide.

For more information about network interfaces that support bypass mode, see the operating manuals for the HSC model.

Using the **Add bridge** button, the administrator can combine several physical interfaces into a new type of interface, a bridge. Provide the following settings:

Name	Description
Enabled	Enables the interface bridge.
Name	The interface name.
Node name	The DCFW cluster node on which the bridge interface is being created.
Type	Specify the interface type as Layer 3 or Layer 2.
Zone	The zone to which the interface bridge belongs.
Netflow profile	The Netflow profile to send statistical data to the Netflow collector. You can read about Netflow profiles in the Netflow Profiles chapter.

Name	Description
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
Bridge interfaces	The two interfaces that will be used to build the bridge.
Bypass bridge interfaces	The interface pair that will be used to build a bypass bridge. DCFW HSC support is required.
STP (Spanning Tree Protocol)	Enables the use of STP to prevent network loops.
Forward delay	The delay before the bridge switches to the active (forwarding) mode if STP is enabled.
Maximum age	The time after which an STP connection is considered lost.
Networking	The IP address assignment method: no address, a static IP address, or a dynamic IP address obtained using DHCP. The capability to change a MAC address, MTU size, and MSS size.
DHCP relay	This is used to configure DHCP relay for the bridge interface. Enable DHCP relay, enter the IP address of the interface on which the relay function is added in the **UserGate address** field, and specify one or more DHCP servers where client DHCP requests are to be forwarded.

PPPoE Interface

PPPoE (**Point-to-point protocol over Ethernet**) is a link-layer network protocol for PPP frame transmission via Ethernet. Using the **Add** button, the administrator can create a PPPoE interface by selecting **Add PPPoE**. To create the interface, provide the following settings:

Name	Description
Enabled	Enables the PPPoE interface.
Node name	The DCFW cluster node on which the PPPoE interface is being created.
Interface	Specify the network interface on which the PPPoE interface will be created.
Zone	The zone to which the PPPoE interface belongs.

Name	Description
Netflow profile	The Netflow profile to send statistical data to the Netflow collector. You can read about Netflow profiles in the Netflow Profiles chapter.
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
MTU	The MTU size. Set by default to a value of 1492 bytes compatible with the standard Ethernet frame size.
MSS	**The MSS size. Correct values are 0, or from 4 to the specified MTU value minus 40.**
Login name	The username for the PPPoE connection.
Password	The password for the PPPoE connection.
Persist connection	Enables automatic reconnection on connection loss.
Authentication type	The authentication protocols used in PPP: • **CHAP**: Challenge Handshake Authentication Protocol, an authentication protocol (algorithm) with three-way handshaking. It avoids transmitting the user password itself by sending certain derived information instead. • **PAP**: Password Authentication Protocol, a simple authentication protocol that involves transmitting the username and password to the remote access server in plain text (without encryption).
Holdoff interval (sec.)	The time interval in seconds before re-connecting on a connection loss.
Default route	Sets the PPPoE interface as the default route.
LCP echo interval (sec.)	The time interval between periodic connection checks.
Number of LCP echo failures	The number of LCP echo failures that, when reached, makes DCFW consider the connection lost and terminate it.
Use provider's DNS	If this option is enabled, DCFW uses DNS servers granted by your provider.
Number of connection attempts	The number of failed connection attempts after which the automatic retries will stop.

Name	Description
PPPoE service	The service name should be specified here if given to you by the provider. If a service name is not used, the field should be left empty.

VPN Interface

A VPN interface is a virtual network adapter that will be used to connect VPN clients. This is a cluster-type interface, which means it will be created automatically on all DCFW nodes included in a configuration cluster. If an HA cluster exists, in case any problems are identified with the active server, VPN clients will be automatically switched to a backup server, and without terminating existing VPN connections.

In the **Network → Interfaces** section, click **Add** and select **Add VPN**. Provide the following settings:

Name	Description
Name	The interface name. Should be in the form of tunnelN, where N is the ordinal number of the VPN interface.
Description	Interface description.
Zone	The zone to which this interface will belong. All clients with a VPN connection to DCFW will be placed in the same zone.
Netflow profile	The Netflow profile to send statistical data to the Netflow collector. You can read about Netflow profiles in the Netflow Profiles chapter.
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
Aggregation mode	The IP address assignment type. The options are no address, a static IP address, or a dynamic IP address obtained using DHCP. If the interface is to be used for accepting VPN connections (Site-2-Site VPN or Remote access VPN), a static IP address must be used. To use an interface as a client, select the dynamic mode.
MTU	The MTU size for the selected interface.
MSS	The MSS size. Correct values are 0, or from 4 to the specified MTU value minus 40.

The system has three predefined VPN interfaces by default:

- **tunnel1**, recommended for a Remote access VPN
- **tunnel2**, recommended for the server side of a Site-to-Site VPN
- **tunnel3**, recommended for the client side of a Site-to-Site VPN.

Tunnel Interface

A tunnel interface is a virtual network adapter that can be used to create a point-to-point connection via an IP network. The following types of tunnel interfaces are supported:

- GRE: a network packet tunneling protocol developed by Cisco Systems. Its main purpose is to encapsulate network layer packets into IP packets. The IP protocol number is 47.
- IPIP: an IP tunneling protocol that encapsulates an IP packet into another IP packet Encapsulating one IP packet in another IP packet adds an external header with Source IP which is the entry point into the tunnel, and Destination IP which is the exit point from the tunnel.
- VXLAN: a protocol for tunneling Layer 2 Ethernet frames into UDP packets. Uses port 4789.

To create a tunnel interface, in the **Network → Interfaces** section, click **Add** and select **Add tunnel**. Provide the following settings:

Name	Description
Enabled	Enable or disable the interface.
Name	The interface name. Should be in the form greN, where N is the ordinal number of the tunnel interface.
Description	Interface description.
Zone	The zone to which this interface will belong.
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
Aggregation mode	The tunnel's operating mode: GRE, IPIP, or VXLAN.

Name	Description
MTU	The MTU size for the selected interface.
MSS	The MSS size (available since software release 7.3.x). Correct values: • 0; • from 4 to the entered MTU value minus 40 (for VXLAN); • from 4 to the entered MTU value minus 60 (for IPIP); • from 4 to the entered MTU value minus 64 (for GRE).
Local IP	The local address of the point-to-point interface.
Remote IP	The remote address of the point-to-point interface.
Interface IP	The IP address assigned to the tunnel interface.
VXLAN ID	The VXLAN ID. Relevant only for a VXLAN tunnel.

Loopback Interface

To create a loopback interface, in the **Network → Interfaces** section, click **Add** and select **Add loopback interface**. Provide the following settings:

Parameter	Description
Enabled	Enables the interface.
Name	Interface name in the loopbackN form, where N is an integer.
Description	An optional interface description.
Node name	Select an DCFW cluster node where the interface is created.
Type	Specify the interface type as Layer 3 or Layer 2.
Zone	The zone to which the interface belongs.
Netflow profile	The Netflow profile to send statistical data to the Netflow collector. You can read about Netflow profiles in the Netflow Profiles chapter.
LLDP profile	LLDP profile to send data using Link Layer Discovery Protocol (LLDP).

Parameter	Description
Alias	An alternative interface name assigned by the administrator. This optional setting is used for working with SNMP. The value is a string with a length of up to 64 characters. **Important!** Cyrillic characters are not allowed in the value.
Networking	The IP address assignment method: no address, a static IP address, or a dynamic IP address obtained using DHCP. The capability to change a MAC address, MTU size, and MSS size.
DHCP relay	Settings for the DHCP relay on the interface. Enable DHCP relay, enter the IP address of the interface on which the relay function is added in the **UserGate address** field, and specify one or more DHCP servers where client DHCP requests are to be forwarded.

Gateway Configuration

To connect DCFW to the Internet, you need to specify the IP address(es) of one or more gateways. If connections to several Internet providers are used, several gateways must be specified. The gateway setting is specific to each cluster node.

Here is an example of a network configuration with two providers:

- Interface eth1 with an IP address of 192.168.11.2 is connected to Internet Provider 1. To enable Internet access via this provider, a gateway with an IP address of 192.168.11.1 must be added.
- Interface eth2 with an IP address of 192.168.12.2 is connected to Internet Provider 2. To enable Internet access via this provider, a gateway with an IP address of 192.168.12.1 must be added

When two or more gateways exist, there are two options:

Name	Description
Traffic load balancing between gateways	Set the **Balancing** checkbox and assign a **Weight** to each gateway. In this case, all traffic destined for the Internet will be distributed between the gateways according to the weights assigned (the greater the weight, the larger portion of the traffic will pass through the gateway). When traffic is distributed between gateways with unequal weights, the following happens: 1. A hash of the source and destination addresses is computed.

Name	Description
	2. A gateway is selected The traffic is distributed based on the weights. Assume that 2 gateways are configured, and: • **n1**, **n2** are the sessions that pass through the gateways; • **w1**, **w2** are the gateway weights. Then the sessions will be distributed between the gateways according to the formula **n1/w1 = n2/w2**.
Main gateway with failover	Select one of the gateways as the main and configure the **Connectivity checker** by clicking the button with that name. The connectivity checker periodically verifies if the host is accessible from the Internet (using ping) with the interval specified in the settings and, if the host ceases to be reachable, switches all traffic to the backup gateways in the order they are listed in the console (if the order has not changed in the current session sorting of displayed gateways; changing the sorting order does not affect the gateway selection process). By default, the network connectivity checker is configured to use Google's public DNS server (8.8.8.8), but this can be changed to any other host if the administrator so desires.

A gateway's status (green for available, red for unavailable) is determined as follows:

Name	Description
Connectivity checker disabled	A gateway is considered available if DCFW can obtain its MAC address using an ARP request. Internet connectivity is not checked for this gateway. If it is not possible to determine the gateway's MAC address, it is considered unavailable.
Connectivity checker enabled	A gateway is considered available if: • DCFW can obtain its MAC address using an ARP request. • Internet connectivity check for this gateway was successful. Otherwise, the gateway is considered unavailable.

DHCP Configuration

The DHCP (Dynamic Host Configuration Protocol) service enables you to automate the process of assigning network settings to clients in the local network. In a network with a DHCP server, each network device can be dynamically assigned an IP address, gateway address, and DNS.

DCFW can also function as a DHCP relay by forwarding DHCP requests from clients located in different networks to a central DHCP server. For more details on configuring DHCP relay, see the Network Interface Configuration section.

In DCFW, you can create several IP address ranges to be assigned by DHCP. DHCP runs independently on each HA cluster node. To ensure the high availability of the DHCP service in a cluster, DHCP should be configured on both nodes with non-overlapping IP address ranges.

To add a DHCP range, click **Add** and provide these settings:

Name	Description
Enabled	Enables or disables the use of this DHCP range.
Node	The cluster node on which the range is being created.
Interface	Interface of the server which will assign IP addresses from the range being created.
IP range	The IP address range assigned to DHCP clients.
Mask	The subnet mask assigned to DHCP clients.
Lease time	The duration in seconds for which IP addresses are assigned.
Domain	The domain name assigned to DHCP clients.
Gateway	The gateway IP address assigned to DHCP clients.
Name servers	The DNS server IP addresses assigned to DHCP clients.
Reserved hosts	The MAC addresses and the associated IP addresses.
Ignored MAC	List of MAC addresses ignored by the DHCP server.
DHCP PXE boot	The server address and boot file name returned in response to a PXE boot request.
DHCP options	

Name	Description
	Option number and value. For the list of available options, see DHCP Options.

The assigned IP addresses are displayed in the **Addresses** pane. The administrator can release any leased IP address by selecting it and clicking **Release**.

> **Note**
> **For DHCP address leasing to work on an interface that resides in a zone with IP spoofing protection enabled, go to the IP spoofing protection tab and specify the IP lease ranges in the zone properties as well as the 0.0.0.0 address.**

DNS Configuration

This section describes how to configure the DNS and DNS proxy services.

In order the product to work correctly, DCFW must be able to resolve domain names into IP addresses. Specify valid IP addresses of DNS servers in the **System DNS servers** setting.

The DNS proxy service enables user DNS requests to be intercepted and modified according to the administrator's needs. This service works both in the explicit mode and for intercepting transit requests. For the explicit mode, DNS access must be allowed in the relevant zone. For intercepting transit requests in this zone, the following DNS proxy settings need to be configured.

These are the DNS proxy settings:

Name	Description
DNS caching	Enables or disables DNS response caching. It is recommended to leave this enabled to speed up client service.
DNS Filtering	Enables or disables DNS request filtering. When DNS filtering is enabled, DCFW checks and intercepts requests, passing them along from its own IP address. If the request matches a content filtering deny rule, it will be blocked. For the filtering to work, you need to purchase a license for the ATP module.

Name	Description
	Important! **The DNS filtering and L2 bridge functionality are not compatible in the current version: when DNS filtering is enabled, DNS requests stop passing through the bridge.**
Recursive DNS queries	Enables or disables recursive DNS queries from the server. It is recommended to leave this enabled.
Max TTL for DNS records (sec)	Sets the maximum possible time to live (TTL) for DNS records.
Limit DNS requests per second for user	Sets a limit for the number of DNS requests per second for each user. Requests in excess of this limit parameter will be rejected. The default value is 100 requests per second. Large values are not recommended for this parameter, because DNS flood (DNS DoS) attacks are a fairly common reason why DNS servers deny service.
Only A and AAAA DNS-records for unknown users (prohibit VPN over DNS)	When this protection is enabled, UserGate will only respond to unknown users if they request A or AAAA records. This effectively blocks attempts to establish a VPN over the DNS protocol.

You can use DNS proxy rules to specify the DNS servers to which requests for certain domains should be forwarded. This option can be useful when your company uses a local domain that is permanently disconnected from the Internet and used for company-internal needs, such as an Active Directory domain.

To create a DNS proxy rule, follow these steps:

Name	Description
Step 1. Add a rule.	Click **Add** and provide a **Name** and an optional **Description**.
Step 2. Specify a domain list.	List the domains that need forwarding, e.g., localdomain.local. "*" can be used to specify a domain template.
Step 3. Specify DNS servers.	List the IP addresses of DNS servers to which the requests for the above domains should be forwarded.

You can also use a DNS proxy to define static host-type records, or A records. To define a static record, follow these steps:

Name	Description
Step 1. Add a record.	Click **Add** and provide a **Name** and an optional **Description**.
Step 2. Specify FQDN.	Enter the **Fully Qualified Domain Name** (FQDN) of the static record, such as www.example.com.
Step 3. Specify IP addresses.	Specify the list of IP addresses that DCFW will return when this FQDN is requested.

Virtual Routers

In large networks, it often happens that multiple logical networks use the same network devices for their traffic. This traffic needs to be separated at the devices, first and foremost to reduce the risk of unauthorized cross-network access.

Virtual routers, or **Virtual Routing and Forwarding (VRF)** features, provide traffic separation by organizing network interfaces into independent groups. The traffic from one interface group cannot reach other interface groups.

Each virtual router has its own routing table. A routing table may contain a record of routes defined statically or obtained using dynamic routing protocols, such as BGP, OSPF, or RIP.

Different virtual routers are allowed to use the same IP networks (IP overlapping).

Network interfaces that have not been assigned explicitly to one of the virtual routers are automatically assigned to the **Default virtual router**.

Virtual routers have the following limitations:

- These services can only be used in the default virtual router:
- WCCP
- ICAP
- DNS
- Authorization

- Any network traffic that is generated by the device itself, such as license checks, update downloads, log uploads, sending email/SMS messages, SNMP traps, etc.
- The NAT, DNAT, and port forwarding rules apply to all virtual routers.
- The zones are global — that is, the zone settings and interface-to-zone mappings apply to all virtual routers.

> **Note**
> **The default virtual router is required for correct operation of DCFW. It is used to check licenses, download updates, and provide DNS services.**

To add a virtual router, follow these steps:

> **Note!**
> **These prefixes cannot be used in the name of a virtual router: port, gre, egress, ingress, tun, tap, erspan, ppp, bond, bridge, pimreg.**

> **Note**
> **When creating a virtual router, its name must not contain capital letters and must be at least three characters long.**

Name	Description
Step 1. Create a new virtual router.	In the **Network → Virtual routers** section, click "Add" and provide a name and description for the new virtual router. Specify the name of the cluster node on which this virtual router is being created, if you have a cluster.
Step 2. Add network interfaces to the newly created virtual router.	On the **Interfaces** tab, select the network interfaces that should be added to this virtual router. Interfaces that are already added to other virtual routers are not available for selection; any single interface can only belong to one virtual router. All types of interfaces, including physical, virtual (VLAN), bond, VPN, and others can be added to a virtual router.
Step 3. (Optional) Add static routes.	Add the routes (except the default route) that will be applied to the traffic in this virtual router. For more details, see the Static Routes section.

Name	Description
	The default route is added in the **Network → Gateways** section. For more details on configuring gateways, see the section Gateway Configuration.
Step 4. (Optional) Add dynamic routes obtained using the OSPF routing protocol.	Configure the OSPF protocol to build a dynamic route map. For more details, see the section OSPF.
Step 5. (Optional) Add dynamic routes obtained using the BGP routing protocol.	Configure the BGP protocol to build a dynamic route map. For more details, see the BGP section.
Step 6. (Optional) Add dynamic routes obtained using the RIP routing protocol.	Configure the RIP protocol to build a dynamic route map. For more details, see the RIP section.
Step 7. (Optional) Configure multicasting.	Configure the multicasting settings for this virtual router. For more details, see the Multicasting section.

Static Routes

This section describes how to specify a route to a network that is behind a specific router. For example, a local network can have a router that combines several IP subnets. The route is applied locally to the specific cluster node and virtual router where it is created.

To add a route, follow these steps:

Name	Description
Step 1. Select a virtual router.	If there are several virtual routers, select the desired one.
Step 2. Provide a name and description for the route.	In the **Network → Virtual routers** section, select **Static routes** in the menu and click **Add**. Provide a name for the new route. Optionally, you can also provide a description for the route.
Step 3. Select the route type.	The following route types are available: • Unicast: the standard route type. Forwards the traffic destined for the specified address via the specified gateway. • Blackhole: drops the traffic without informing the source that the data did not reach the recipient.

Name	Description
	• Unreachable: drops the traffic. and sends the "Host unreachable" (type 3 code 1) ICMP message to the source. • Prohibit: drops the traffic. and sends the "Host unreachable" (type 3 code 13) ICMP message to the source.
Step 4. Specify the destination address.	Specify the subnet where the route will point to, such as 172.16.20.0/24 or 172.16.20.5/32.
Step 5. Specify the gateway.	Specify the IP address of the gateway through which the above subnet will be accessible. This IP address must be reachable from DCFW.
Step 6. Specify the network interface.	Specify the network interface through which the route will be added. If you keep the default value, **Automatically**, DCFW will determine the interface based on the IP address settings of the available network interfaces.
Step 7. Specify the metric.	Specify the metric for the route. The lower the metric value, the higher the route's priority, if there are multiple routes to this network.

Dynamic Routing Protocols

Dynamic routing protocols are used to signal which networks are currently connected to each of the routers. Routers communicate using routing protocols. DCFW updates the kernel routing table in accordance with the information it receives from neighboring routers.

Dynamic routing does not change how the kernel performs routing at the IP layer. The kernel keeps looking up routes to hosts and networks as well as default routes in its routing table. The only thing that changes is how routes are managed in the routing table: instead of the manual method, they are added and removed dynamically.

> **Note**
> **If static gateways are configured in the system, the default routes obtained using dynamic routing protocols are ignored.**

DCFW supports three routing protocols: OSPF, BGP, and RIP.

OSPF

Dynamic routing protocols are used to signal which networks are currently connected to each of the routers. Routers communicate using routing protocols. DCFW updates the kernel routing table in accordance with the information it receives from neighboring routers. Dynamic routing does not change how the kernel performs routing at the IP layer. The kernel keeps looking up routes to hosts and networks as well as default routes in its routing table. The only thing that changes is how routes are managed in the routing table: instead of the manual method, they are added and removed dynamically. Routes are only added to the virtual router in which the OSPF protocol is configured.

OSPF (Open Shortest Path First) is a dynamic routing protocol based on the link-state monitoring technology and using Dijkstra's algorithm to find the shortest path.

The OSPF protocol disseminates information on the available routes among the routers that operate within a single autonomous system (AS). For more details on how the OSPF protocol works, see the relevant technical documentation.

> **Note**
> **When OSPF is used in an Active-Passive HA cluster, a node with the slave role automatically assigns a cost to all its interfaces and redistribution lists that is twice as high as that set on the node. This ensures that the master node has the priority in traffic routing.**

To configure OSPF in DCFW, follow these steps:

Name	Description
Step 1. Select a virtual router.	If there are several virtual routers, select the desired one.
Step 2. Enable the OSPF router.	In the DCFW console, go to the **Network → Virtual routers** section, select **OSPF** in the menu, and configure the OSPF router.

To configure an OSPF router, provide the following settings:

Name	Description
Enabled	Enables or disables this OSPF router.
Router ID	The router's IP address. Must be unique and set up in IPv4 format (for the sake of convenience, it may match one of the IP

Name	Description
	addresses assigned to DCFW network interfaces that belong to this virtual router).
Redistribute	Distribute routes to networks directly connected to DCFW (connected) or static routes added by the administrator for this virtual router (kernel) to other OSPF routers.
Metric	Set a metric for the distributed routes. To set the default metric, enter 0 in this field. (The default metric for the cluster master node is 20. The default metric for the backup node is 40.)
Default originate	Notify other routers that this router has a default route.

To configure OSPF interfaces, provide these settings:

Name	Description
Enabled	Enable or disable the interface.
Interface	Select one of the existing interfaces on which OSPF will run. Only the interfaces belonging to this virtual router are available for selection.
Network type	Select a network type to optimize the adjacency establishment process. The following settings are available: • **Not specified**. • **Broadcast**. • **Point-to-point**. • **Point-to-multipoint**.
Passive mode	Enable/disable the passive operating mode of the interface, in which routing protocol update packets are prohibited from being sent through the interface.
Cost	The link cost for this interface. This value is reported in the LSA (link-state advertisement) to the neighboring routers which use it to compute the shortest path. Default value: 1.
Priority	An integer in the range from 0 to 255. The higher the value, the higher the probability that this router will become the network's designated router for sending out LSAs. A value of 0 excludes the router from being designated. Default value: 1.
Hello interval	

Name	Description
	The time interval in seconds between hello packets sent by the router. This should be the same for all routers in an autonomous system. The default value is 10 seconds.
Dead interval	The time interval in seconds after which the neighboring router is considered offline. The time is counted from the moment of receiving the last hello packet from the neighboring router. The default value is 40 seconds.
Retransmit interval	The time interval before LSA packet retransmission. The default value is 5 seconds.
Transmit delay	The approximate time it takes to deliver a link state update to the neighboring routers. The default value is 1 second.
Bfd profile	Defines BFD settings for OSPF monitoring. This makes it possible for the corresponding BFD session connection events to instantly update the OSPF interface status. For more details, see the BFD Profiles section.
Authentication Enabled	Turns on mandatory authentication for each OSPF message received by the router. Authentication is normally used to prevent the injection of a fake route from illegitimate routers.
Authentication type	The options are: • **Plain**: send the key in plain text for router authentication. A value must be provided for the **Key** field. • **Digest**: use an MD5 hash of the key to authenticate OSPF packets. The values of **Key** and **MD5 key ID** must be provided. For authentication to work correctly, these parameters must be identical on all routers. The Key value can only include Latin letters, numbers, and the underscore character. Maximum length: 16 characters.

To configure OSPF areas, provide these settings:

Name	Description
Enabled	Enables or disables this area.
Name	The area name.
Cost	The cost of the default route advertised to the stub area. The default value is 1. In the case where there are multiple ABRs between a stub area and another area, the administrator can assign different costs

Name	Description
	advertised from the ABRs to the stub area to prioritize traffic from the stub area through one of those ABRs.
Area ID	The ID for the area. The ID can be specified in decimal format or IP address record format. The area ID must match to establish an OSPF adjacency.
Authorization type	The options are: • **None**: do not require OSPF packet authorization. • **Plain**: transmit the key as plain text to authenticate OSPF packets. The key specified in the interface settings is used. • **Digest**: use an MD5 hash of the key to authenticate OSPF packets. The key specified in the interface settings is used. The interface-level authentication takes precedence over zone-level authorization.
Area type	Defines the type of the area. The following area types are supported: • **Normal**: a normal area created by default. This zone receives link updates, summary routes, and external routes. • **Stub**: a stub area. Does not receive information on routes external to the autonomous system but receives routes from other areas. If routers from a stub area need to send information outside of the autonomous system, they use the default route. An ASBR cannot reside in a stub area. • **NSSA**: Not-so-stubby. A NSSA area defines an additional type of LSA, LSA type 7. A boundary router (ASBR) can be located in the NSSA zone.
No summary	Prohibits injecting summarized routes into stub-type areas.
Interfaces	Select the OSPF interfaces on which this area will be available.
Virtual links	This is a special type of connection that makes it possible, for example, to interconnect a partitioned area or connect an area to the backbone area via another area. It is configured between two ABRs. Routers can transmit OSPF packets encapsulated in IP packets over such links. This mechanism is used as a temporary solution or as a backup in case the primary connections fail. You can specify the IDs of the routers available via this zone.

BGP

Dynamic routing protocols are used to signal which networks are currently connected to each of the routers. Routers communicate using routing protocols. DCFW updates the kernel routing table in accordance with the information it receives from neighboring routers. Dynamic routing does not change how the kernel performs routing at the IP layer. The kernel keeps looking up routes to hosts and networks as well as default routes in its routing table. The only thing that changes is how routes are managed in the routing table: instead of the manual method, they are added and removed dynamically. Routes are only added to the virtual router in which the BGP protocol is configured.

BGP (Border Gateway Protocol) is a dynamic routing protocol classified as an External Gateway Protocol (EGP). EGP — External Gateway Protocol). Currently, it is the main dynamic routingprotocol used on the Internet. The BGP protocol is designed to exchange routing and reachability information among autonomous systems (AS), which are groups of routers with common technical management and administration that use intra-domain routing protocols to determine routes within a group and an inter-domain routing protocol to determine routes for packet delivery to other ASs. The information transmitted includes the list of ASs that can be accessed via this system. The best routes are selected based on the rules that are in place in the network. For more details on how the BGP protocol works, see the relevant technical documentation.

To configure BGP in DCFW, follow these steps:

Name	Description
Step 1. Select a virtual router.	If there are several virtual routers, select the desired one.
Step 2. Enable the BGP router.	In the DCFW console, go to the **Network → Virtual routers** section, select **BGP** in the menu, and configure the BGP router.
Step 3. Specify the filters and optional routemaps to limit the number of routes to receive.	In the **Filters** section, click **Add** and configure the Routemap and filter settings. Add as many routemaps/filters as required for BGP to work in your organization.
Step 4. Add at least one BGP neighbor (peer).	In the **Neighbors** section, click **Add** and configure the router settings for the neighboring AS. Add as many neighbors as required. **Important!** RFC 8212 includes a mandatory requirement that export and import filters be added for each neighbor. Without import filters, the router will not receive routes from that neighbor, and without export filters, the router will not advertise routes to that neighbor.

Name	Description
	If several IP addresses are assigned to the DCFW interface from which the connection to a neighbor is being established, then in absence of a NAT rule that force-assigns a source address to the BGP session with this neighbor, you need to specify the primary IP address (i.e., the one listed first in the interface settings) as the DCFW address when configuring a BGP neighbor.

To configure a BGP router, provide the following settings:

Name	Description
Enabled	Enables or disables this BGP router.
Router ID	The router's IP address. Must match one of the IP addresses assigned to DCFW network interfaces that belong to this virtual router.
AS number	An autonomous system is a system of IP networks and routers managed by one or more operators that have a single routing policy. The autonomous system number identifies the router as belonging to that system.
Redistribute	Lets other BGP routers to distribute routes to networks directly connected to DCFW (connected), static routes added by the administrator for this virtual router (kernel), or routes received via the OSPF protocol.
Multiple path	Enables traffic load balancing to routes with identical cost.
Networks	The list of networks that belong to this AS.

Note

If you enter networks in the *Networks* section that are not in the routing table, they will not be advertised.

To add BGP neighbors, click **Add** and provide these settings:

Name	Description
Enabled	Enables or disables this neighbor.
Host	The neighbor's IP address.

Name	Description
Description	An arbitrary description for the neighbor.
Remote ASN	The neighbor's AS number.
Weight	The weight assigned to route data received from this neighbor.
TTL	The maximum allowed number of hops to this neighbor.
Bfd profile	Configure BGP monitoring using the BFD profile to enable faster detection of connection faults. For more information on configuring BFD, see BFD Profiles.
Announce self as next hop for BGP	Replace the next-hop-self value with own IP address, if the neighbor uses BGP.
Multihop for eBGP	Indicates that the connection to this neighbor is indirect (more than a single hop).
Route reflector client	Indicates if the neighbor is a route reflector client.
Soft reconfiguration	Use soft reconfiguration (without terminating connections) for configuration updates.
Default originate	Advertise the default route to this neighbor.
Authentication	Enables authentication for this neighbor. The authentication password is set here.
BGP neighbor filters	Limits the route information received from the neighbors or advertised to them.
Routemaps	Routemaps are used to manage routing tables and specify the match conditions under which routes are passed between domains.

A routemap allows filtering of routes on redistribution and modification of various route attributes. To create a routemap, provide the following settings:

Name	Description
Name	The routemap name.

Name	Description
Action	Sets the action for this routemap. Can take the following values: • **Allow**: allows data that matches the routemap conditions. • **Block**: blocks data that matches the routemap conditions.
Match by	Routemap conditions. Can take the following values: • **IP** If this condition is selected, go to the IP addresses tab and add all required IP addresses for the condition. • **AS path**. If this condition is selected, go to the AS path tab and add all required AS numbers for the condition. POSIX 1003.2 regular expressions are allowed, supplemented by the underscore (_) character that is interpreted as: • A space • A comma • Start of line • End of line • AS set delimiter { and } • AS confederation delimiter (and). • **Community**. If this condition is selected, go to the Community tab and add all required BGP community strings for the condition.
Set next hop	Set the next hop value for the filtered routes to this IP address.
Set weight	Set the weight for the filtered routes to this value.
Set metric	Set the metric for the filtered routes to this value.
Set preference	Set the preference for the filtered routes to this value.
Set AS prepend	Set the AS-prepend value, which is a list of autonomous systems added for this route.
Community	Set the BGP community value for the filtered routes.

Filters allow you to filter routes when redistributing. To create a filter, provide the following settings:

Name	Description
Name	The filter name.
Action	Sets the action for this filter. Can take the following values: • **Allow**: allows data that matches the filter conditions. • **Block**: blocks data that matches the filter conditions.
Filter by	Filter conditions. Can take the following values: • **IP** If this condition is selected, go to the **IP addresses** tab and add all required IP addresses for the condition. The addresses can be specified in the following formats: ◦ 10.0.0.0/8 for the 10.0.0.0/8 subnet only ◦ 10.0.0.0/8::11 for routes where the first octet is 10 and the prefix is from 8 to 11 ◦ 10.0.0.0/8:11:13 for routes where the first octet is 10 and the prefix is from 11 to 13. • **AS path**. If this condition is selected, go to the **AS path** tab and add all required AS numbers for the condition.

RIP

Dynamic routing protocols are used to signal which networks are currently connected to each of the routers. Routers communicate using routing protocols. DCFW updates the kernel routing table in accordance with the information it receives from neighboring routers. Dynamic routing does not change how the kernel performs routing at the IP layer. The kernel keeps looking up routes to hosts and networks as well as default routes in its routing table. The only thing that changes is how routes are managed in the routing table: instead of the manual method, they are added and removed dynamically. Routes are only added to the virtual router in which the RIP protocol is configured.

RIP (Routing Information Protocol) is a distance-vector routing protocol that uses intermediate sections (hops) as a routing metric. For more details on how the RIP protocol works, see the relevant technical documentation.

To configure RIP in DCFW, follow these steps:

Name	Description
Step 1. Select a virtual router.	If there are several virtual routers, select the desired one.

Name	Description
Step 2. Enable the RIP router.	In the DCFW console, go to the **Network → Virtual routers** section, select **RIP** in the menu, and configure the RIP router.
Step 3. Specify the RIP networks.	In the DCFW console, go to the **Network → Virtual routers** section, select **RIP** in the menu, and specify RIP networks the RIP protocol will be used with.
Step 4. Configure the RIP interfaces.	In the DCFW console, go to the **Network → Virtual routers** section, select **RIP** in the menu, and configure the RIP interfaces.

To configure an RIP router, provide the following settings:

Name	Description
Enabled	Enables or disables this RIP router.
RIP version	Specifies the RIP protocol version. Normally, v2 is used.
Default metric	The route cost. The metric is normally equal to 1 and cannot exceed 15.
Administrative distance	The cost of routes received using the RIP protocol. Default value for RIP protocol: 120. This is used for route selection when routes can be received using multiple methods (OSPF, BGP, static).
Default originate	Notify other routers that this router has a default route.

A RIP router will send routing updates only from the interfaces for which **RIP networks** are specified. At least one network must be specified for the protocol to work correctly. The administrator can specify the RIP network using the CIDR notation, such as 192.168.1.0/24, or select the network interface from which updates will be sent.

To configure RIP interfaces, provide these settings:

Name	Description
Interface	Select the interface that will be used for RIP routing. Only the interfaces belonging to this virtual router are available for selection.
Send version	Specify the RIP protocol version that the router will send.

Name	Description
Receive version	Specify the RIP protocol version that the router will receive.
Password	The authorization string that will be sent and received in RIP packets. All routes participating in RIP information exchange must have an identical password.
Split horizon	A method of preventing routing loops where the router does not send network information via the interface on which the update was received.
Poison reverse	A method of preventing routing loops where the router sets a route cost of 16 and sends it to the neighbor from which it was received.
Passive mode	Sets an operating mode where the interface receives RIP updates but does not send them.

In the route redistribution settings, you can specify which routes need to be sent to the neighbors. Redistribution can be enabled for routes received using the OSPF and BGPG dynamic routing protocols, routes directly connected to the DCFW network (connected), and routes added by the administrator in the **Routes** section (kernel).

Multicasting

The IP multicast technology enables a significant reduction in the amount of network traffic by delivering a single information stream to thousands and even larger numbers of consumers, which is especially efficient for voice and video traffic delivery. The traditional traffic delivery methods are unicast (point-to-point) and broadcast. Multicast allows delivery of traffic to a group of hosts, called a multicast group. The recipient hosts that want to receive this traffic must join (become members of) the corresponding multicast group. To add hosts to a multicast group, the Internet Group Management Protocol (IGMP) is used. A multicast group is identified by its multicast address. For multicast addresses, a Class D subnet is reserved with the most significant 4 bits set to 1110. Thus, the address range for multicasting is defined as 224.0.0.0 — 239.255.255.255.

Routers need to provide efficient traffic delivery from the multicast source to the recipients. For that purpose, the Protocol Independent Multicast (PIM) is used in routers.

Routers in a multicast environment can have one of the three roles: First Hop Router (FHR), Rendezvous Point (RP), and Last Hop Router (LHR). The FHR is located closest to the multicast source and is responsible for registering the source in the network. The RP is a catalog of available multicast sources for the Any Source Multicast (ASM)

mode. The LHR is located closest to the multicast recipient. Clients (multicast recipients) in local networks connected to the LHR use the IGMP protocol to register in the multicast group of interest by sending an IGMP membership report message.

DCFW can be used as an LHR for the local networks connected to it. For client (recipient) registration, DCFW supports the IGMPv3 and IGMPv2 protocols.

For communicating with other multicast routers, DCFW can only use the PIM Sparse Mode (PIM-SM). This is a mode where multicast traffic is sent only to those recipients that have explicitly requested it. The recipients must periodically confirm their desire to receive multicast traffic.

DCFW supports Source Specific Multicast (SSM) and Any Source Multicast (ASM) modes.

Source Specific Multicast (SSM) is used when the recipient of the traffic explicitly specifies a multicast source known to it. In this mode, addresses are written as follows:

rtp://<src_ip>@<group_address>:<port>, where src_ip is the multicast source address, group_address is the multicast group address, and port is the port. Example: rtp://10.10.10.10@239.0.0.5:4344

In Any Source Multicast (ASM) mode, the multicast recipient specifies the multicast group from which it wants to receive multicast traffic. For this mode to work, a Rendezvous Point (RP) router is required. The RP determines the multicast source for this multicast group and this recipient, and then the source and recipient choose the best network path for sending this multicast traffic. In this mode, addresses are written as follows:

rtp://@<group_address>:<port>, where group_address is the multicast group address and port is the port. Example: rtp://@239.0.0.5:4344

To configure DCFW as an LHR multicast router, follow these steps:

Name	Description
Step 1. Configure a multicast router.	In the DCFW console, go to the **Network → Virtual routers** section, select **Multicast router** in the menu, and configure it.
Step 2. Specify the interfaces on which this router will work.	In the DCFW console, go to the **Network → Virtual routers** section, select **Interfaces** in the menu, and configure the interfaces. Only the interfaces belonging to this virtual router are available for selection.

Name	Description
Step 3. (Optional) Define the Rendezvous points for ASM.	In the DCFW console, go to the **Network → Virtual routers** section, select **Rendezvous points** in the menu, and specify the addresses of the rendezvous points.
Step 4. (Optional) Set the desired restrictions on the available multicast groups for ASM.	In the DCFW console, go to the **Network → Virtual routers** section, select **Rendezvous points** in the menu, and in the **ASM allowed groups** tab specify the addresses of the allowed multicast groups. If you leave the list empty, all multicast group addresses will be allowed.
Step 5. (Optional) Set the desired restrictions on the available multicast groups for SSM.	In the DCFW console, go to the **Network → Virtual routers** section, select **SSM allowed groups** in the menu, and specify the addresses of the allowed multicast groups. If you leave the list empty, all group addresses will be allowed.

When configuring a multicast router, you can provide these settings:

Name	Description
Enabled	Enables or disables the multicast router in this virtual router.
Use ECMP	Enables multi-path traffic distribution using the Equal Cost Multi Path (ECMP) technology. Requires that several routes exist to the network node of interest. If this option is disabled, all traffic to a specific destination host will be sent through only one of the routers (next hop).
Use ECMP rebalance	If this option is enabled and one of the interfaces used for sending traffic has gone offline, all existing streams will be redistributed between the remaining routes (next hop). If disabled, only those streams will be redistributed which were sent via the now-offline interface.
Keep-alive time (sec)	The time interval in seconds (31-60,000) which the router will use to send keepalive messages to neighbors as well as the time to wait before considering the neighbor unavailable.

When configuring interfaces, you can provide these settings:

Name	Description
Enabled	Enables or disables multicasting on this interface.
Interface	Select the interface that will be used for multicasting. Only the interfaces belonging to this virtual router are available for selection.

Name	Description
Multicast HELLO sending timeout (sec)	The time interval in seconds (1-180) used to send PIM HELLO messages. PIM Hello messages are sent periodically from all interfaces for which multicast support is enabled. These messages let the router know about neighbor routers that support multicasting.
DR selection priority	The router's priority (1-4294967295) in the selection of a Designated router (DR). The administrator can use this to manage DR selection for the local network.
Enable IGMP	Receive IGMP report and IGMP query messages on this interface.
Use IGMPv2	Use version 2 of IGMP. By default, version 3 (IGMP v3) is used.

When configuring Rendezvous points, you can specify the following parameters:

Name	Description
Enabled	Enables or disables this RP.
Name	The RP name.
IP address	The unicast IP address of this RP.
Allowed ASM groups	The list of allowed multicast group addresses for any-source multicast from this RP. Any networks in the range 224.0.0.0/4. If empty, there are no restrictions.

Allowed SSM groups: specifies the list of allowed multicast group addresses for source-specific multicast. Any networks from the range 232.0.0.0/8 can be specified. If empty, there are no restrictions.

SPT exclusions: specifies the list of IPv4 multicast groups excluded from switching to the shortest path tree.

WCCP

Web Cache Communication Protocol (WCCP) is a content redirection protocol developed by Cisco. It provides a mechanism for real-time traffic flow redistribution and has native scaling, load balancing, and high availability features. When WCCP is used, the WCCP server receives an HTTP request from a client browser and redirects

it to one or more WCCP clients. A WCCP client receives data from the Internet and returns it to the client browser. The data can be delivered to the client either through the WCCP server or bypassing it, depending on the routing rules.

DCFW can function as a WCCP client. The WCCP server role is normally fulfilled by the router. You can filter traffic received using WCCP using all available filtering mechanisms.

A WCCP service group is a set of WCCP servers (routers, switches) and clients (DCFW) with common traffic redirection settings. The servers in the same service group must have identical settings.

To configure the WCCP client in DCFW, follow these steps:

Name	Description
Step 1. Configure a WCCP server.	Configure a WCCP server according to the instructions given in its documentation.
Step 2. Configure WCCP service groups.	In the DCFW console, go to the **Network → WCCP** section, click **Add**, and create one or more WCCP service groups.

For each service group, provide these settings:

Name	Description
Enabled	Enables or disables this service group.
Name	The service group name.
Description	A description of the service group.
Service group	The numeric ID of the service group. Service group IDs must be identical on all devices in the group.
Priority	The group's priority. If multiple service groups are applicable to the traffic managed by the WCCP server, the priority determines the order in which the server will distribute traffic to the WCCP clients.
Password	The password required to authenticate DCFW in the service group. The password must match the one specified on the WCCP servers.
Forwarding type	Determines the type of forwarding the traffic from WCCP servers to DCFW. The possible values are: • **gre**: use a Generic Routing Encapsulation (GRE) tunnel

Name	Description
	• **L2**: using L2 redirection. In this case, the router (WCCP server) changes the destination MAC address in the packet to the DCFW address. L2 redirection generally requires fewer resources than gre, but the WCCP server and DCFW must reside in the same L2 segment. Not all WCCP server types support L2 redirection with WCCP clients. **Important!** For traffic received via a WCCP tunnel, DCFW will use the client computer's IP address as the source IP, and the source zone will be undefined. Therefore you should not explicitly specify the zone in the source zone filtering rules (leave the "Any" value).
Returning type	Determines the type of forwarding the traffic from DCFW to WCCP servers. The possible values are: • **gre**: use a Generic Routing Encapsulation (GRE) tunnel • **L2**: using L2 redirection. In this case, DCFW (WCCP client) changes the destination MAC address in the packet to the router address (WCCP server). L2 redirection generally requires fewer resources than gre, but the WCCP server and DCFW must reside in the same L2 segment. Not all WCCP server types support L2 redirection with WCCP clients.
Ports to redirect	The ports to redirect. Specify the destination ports for traffic here. If you need to list multiple ports, separate them with a comma, for example: 80, 442, 8080 To redirect traffic based on source port values, you must select the **Source port** checkbox. **Important!** DCFW can only apply filtering to redirected TCP traffic with destination ports 80 and 443 (HTTP/HTTPS). Traffic sent to DCFW through other ports is sent to the Internet unfiltered.
Protocol	Specify the protocol as TCP or UDP.
WCCP routers	Specify the IP addresses of the WCCP servers (routers).
Assignment type	When there are multiple WCCP clients in a service group, the assignment type determines how traffic is distributed from the WCCP servers to the WCCP clients. The available options are: • Hash: distribute traffic based on a hash computed from the specified IP packet fields. Alternate hash: if configured, will be used by the WCCP server on

Name	Description
	exceeding a certain number of packets sent to the WCCP client using the regular hash. The set of IP packet fields used for hashing must be different for the regular and alternate hash. • Mask: distribute traffic based on the result of a Boolean AND between the mask and the selected packet header. When selecting a mask, consult the vendor documentation for the WCCP server.

USERS AND DEVICES

Users and Groups

Security policies, firewall rules, safe browsing rules, and many other features of UserGate DCFW can be applied to users or user groups. The ability to apply policies only to the relevant users gives the administrator the flexibility to configure the network to the organization's requirements.

User identification is a fundamental part of DCFW functionality. A user is considered identified if the system has unambiguously associated the user with the IP address of the device they use to connect to the network. DCFW uses different user identification mechanisms:

- Explicitly defined IP address.
- Login name and password.
- Dedicated terminal server agent (for Microsoft Terminal Server user identification).
- Authorization agent (for Windows systems).
- NTLM or Kerberos protocol.

User identification using name and password can be performed via the captive portal, which in turn can be configured to identify users with the help of Active Directory, RADIUS, TACACS+, NTLM or Kerberos directories or a local user database.

DCFW defines the following user types:

Name	Description
Unknown user	Represents the set of users not identified by the system.
Known user	Represents the set of users identified by the system. The methods of user identification can differ and will be described in more detail later in this chapter.
Any user	This is a union of the **Known** and **Unknown** user sets.
Specific user	A specific user defined and identified in the system; e.g., DOMAIN\User, identified using Active Directory domain authorization.

Users and user groups can be added on a DCFW device itself (these are known as **local users and groups**) or obtained from external directories, such as Microsoft Active Directory.

Groups

User groups enable distinct sets of users to be defined for easier security policy management.

Users

In this section, you can add local users as well as temporarily disable or re-enable them.

The required settings for creating a local user are the username and login. The rest of the settings are optional, but for correct identification, the following must be provided:

- Login and password — for identification using a name and password. In this case, you will need to configure the captive portal where a user can enter their login name and password for authentication.
- IP address, IP address range, or MAC address — for identification using a combination of MAC and IP addresses. Here you need to make sure that the user always accesses the network from the specified MAC and/or IP address.
- VLAN ID — for identification using a VLAN tag. In this case, you need to make sure that the user always accesses the network from the specified VLAN.

- Email: the user's email address. If specified, this can be used to send information, such as a 2nd authentication factor, to the user by email.
- Phones: the user's phone numbers. If specified, this can be used to send information, such as a 2nd authentication factor, to the user by SMS.

If both login/password and IP/MAC/VLAN addresses are specified for the user, the system uses address-based identification. In other words, address-based identification takes priority.

LDAP user accounts are not displayed here, but these users can also be used in security policies.

Auth servers

Authentication servers (auth servers) are external sources of user accounts, such as an LDAP server, or servers that perform authentication for DCFW, such as RADIUS, TACACS+, Kerberos, and SAML. The system supports the following types of authentication servers:

RADIUS, TACACS+, NTLM, and SAML authentication servers can only authenticate users, while a LDAP connector also makes it possible to obtain information on users and their properties.

LDAP Connector

An LDAP connector allows you to:

- Obtain information on users and groups from Active Directory or other LDAP servers. FreeIPA is supported with an LDAP server. The users and groups can be used in filtering rules.
- Authorize users via Active Directory/FreeIPA domains using the captive portal, Kerberos, and NTLM authentication methods.

To create an LDAP connector, click **Add**, select **Add LDAP connector**, and provide the following settings:

Name	Description
Enabled	Enables or disables the use of this authentication server.
Name	The name of the authentication server.

Name	Description
SSL	This specifies whether SSL is required to connect to the LDAP server.
LDAP domain name or IP address	The IP address of the domain controller, the domain controller FQDN or the domain FQDN (e.g., test.local). If the domain controller FQDN is specified, DCFW will obtain the domain controller's address using a DNS request. If the domain FQDN is specified, DCFW will use a backup domain controller if the primary one fails. If some domain controllers are unavailable from the DCFW operation site, you should add a static record to the DNS settings section, where the addresses of available domain controllers were specified, and then use this record's name for the connector.
Bind DN ("login")	The username for connecting to the LDAP server. Must be in the DOMAIN\username or username@domain format. This user must be already created in the domain.
Password	The user's password for connecting to the domain.
LDAP cache lifespan	LDAP cache lifespan (from 1 to 48 hours). The new TTL applies to new entries added to the LDAP cache after the administrator sets it. (Available starting from UGOS version 7.1.3)
LDAP domains	The list of domains served by the specified domain controller, e.g., in case of a domain tree or an Active Directory domain forest. Here you can also specify the short NetBIOS domain name. The domains listed here will be available for selection on the captive portal's auth page if the corresponding option is enabled. For more details on configuring the captive portal, see the Captive Portal Configuration section.
Search roots	The list of LDAP server paths relative to which the system will search for users and groups. Specify the full name, e.g., ou=Office,dc=example,dc=com.
Kerberos keytab	Here you can upload a keytab file for Kerberos authentication. For more details on Kerberos authentication and creating a keytab file, see the Kerberos Authentication Method section.

After creating a server, you should validate the settings by clicking **Check connection**. If your settings are correct, the system will report that; otherwise, it will tell you why it cannot connect.

> **Note**
> **To gain authorization using an LDAP connector, the users must be members of the "Domain users" domain group.**

The LDAP connector configuration is now complete. For LDAP user authorization using a name and password, you need to create captive portal rules. The captive portal is described in more detail in the following chapters.

To add an LDAP user or user group to the filtering rules, click **Add LDAP user/Add LDAP group**, type at least one character present in the names of the desired objects in the search field, and then click **Search** and select the users or groups of interest.

RADIUS User Authentication Server

The RADIUS server option enables user authentication on RADIUS servers, with DCFW acting as a RADIUS client. When authorization is done using a RADIUS server, DCFW sends the username and password information to the RADIUS server, which then responds whether the authentication was successful.

A RADIUS server cannot provide a list of users to DCFW. Therefore, if users were not added to DCFW in advance (e.g., as local users or users fetched from an AD domain using an LDAP connector), only users of **Known** (those who successfully authenticated with the RADIUS server) and **Unknown** (those who were not authorized) types can be used in filtering policies.

To add a RADIUS authentication server, click **Add**, select **Add RADIUS server**, and provide the following settings:

Name	Description
Enabled	Enables or disables the use of this authentication server.
Server Name	The name of the authentication server.
Shared secret	Pre-shared key used by the RADIUS protocol for authentication.
Host	The IP address for the RADIUS server.
Port	The UDP port on which the RADIUS server listens for authentication requests. By default, UDP port 1812 is used.

After adding the authentication server, you need to configure the captive portal for using the RADIUS method. The captive portal is described in more detail in the following chapters.

TACACS+ User Authentication Server

The TACACS+ option enables user authentication on TACACS+ servers. When authorization is done using a TACACS+ server, DCFW sends the username and password information to the server, which then responds as to whether the authentication was successful.

A TACACS+ server cannot provide a list of users to DCFW. Therefore, if users were not added to DCFW in advance (e.g., as local users or users fetched from an AD domain using an LDAP connector), only users of **Known** (those who successfully authenticated with the TACACS+ server) and **Unknown** (those who were not authorized) types can be used in filtering policies.

To add a TACACS+ authentication server, click **Add**, select **Add TACACS+ server**, and provide the following settings:

Name	Description
Enabled	Enables or disables the use of this authentication server.
Server Name	The name of the authentication server.
Secret	Pre-shared key used by the TACACS+ protocol for authentication.
Address	The IP address for the TACACS+ server.
Port	The UDP port on which the TACACS+ server listens for authentication requests. By default, UDP port 1812 is used.
Use single TCP connection	Use a single TCP connection for communicating with the TACACS+ server.
Timeout (sec.)	The authentication timeout for the TACACS+ server. The default is 4 seconds.

SAML IDP User Authentication Server

The SAML IDP (Security Assertion Markup Language Identity Provider) option enables user authorization using a Single Sign-On (SSO) system deployed in the organization, such as Microsoft Active Directory Federation Service. With this method, a user who has been authorized in the SSO system (and not logged out

since) will be transparently authorized on all resources that support SAML authentication. DCFW can be configured as a SAML service provider that uses SAML IDP servers for client authorization.

A SAML IDP server cannot provide a list of local user properties to DCFW. Therefore, if you have not configured an AD domain connection through an LDAP connector, only users of **Known** (those who successfully authenticated with the SAML server) and **Unknown** (those who were not authenticated) types can be used in filtering policies.

To configure authorization using an SAML IDP server, follow these steps:

Name	Description
Step 1. Create DNS records for DCFW.	On the domain controller, create a DNS record that corresponds to DCFW and to be used as an auth.captive domain (e.g., utm.domain.loc). Specify the address of a DCFW interface connected to the **Trusted** network as an IP address.
Step 2. Configure DNS servers in DCFW.	In the DCFW settings, set the domain controller IP addresses as the system DNS servers.
Step 3. Change the **Captive portal auth domain** address.	In the **General settings** section, change the **Captive portal auth domain** address to the DNS record created in the previous step. For more details on changing the captive portal's Auth domain address, see the General Settings section.
Step 4. Configure the SAML IDP server.	On the SAML IDP server, add a record for the DCFW service provider specifying the FQDN name created at Step 1.
Step 5. Create the SAML IDP user authentication server.	Create the SAML IDP user authentication server on DCFW.

To do that, go to the **Users and devices → Auth servers** section, click **Add**, select **Add SAML IDP server** and provide the following settings:

Name	Description
Enabled	Enables or disables the use of this authentication server.
Server Name	The name of the authentication server.
Description	Auth server description.
SAML metadata URL	The URL on the SAML IDP server from where an XML file with a valid configuration for this SAML service provider (client) can be downloaded. When you click **Upload**, the relevant authentication server settings fields will be populated with the

Name	Description
	data from that XML file. This is the preferred method of configuring a SAML IDP authentication server. For more details, see the documentation for the SAML IDP server that you use.
SAML IDP certificate	The certificate that will be used on the SAML client. The available options are: • Create new certificate from downloaded: if the XML upload method was used to configure the server, a new certificate is automatically created and assigned the SAML IDP role (see the Certificate Management section). • Use existing certificate. The certificate must have already been created or imported in the **Certificates** section and must not have a role assigned to it. After you add and save the authentication server, this certificate will be assigned the SAML IDP role. • Do not use certificate.
Single sign-on URL	The URL that is used on the SAML IDP server as the single login point. For more details, see the documentation for your SAML IDP server.
Single sign-on binding	The method used to work with a SSO single login point. Options: **POST** and **Redirect**. For more details, see the documentation for your SAML IDP server.
Single logout URL	The URL used on the SAML IDP server as the single logout point. For more details, see the documentation for your SAML IDP server.
Single logout binding	The method used to work with a SSO single logout point. Options: **POST** and **Redirect**. For more details, see the documentation for your SAML IDP server.

NTLM Authentication Server

The NTLM option enables transparent (i.e., without requesting a username and password) authorization of Active Directory domain users. With NTLM authorization, DCFW works with the domain controllers that authenticate users for Internet access.

An NTLM server cannot provide a list of users to DCFW. Therefore, if users were not added to DCFW in advance (e.g., as local users or users fetched from an AD domain using an LDAP connector), only users of **Known** (those who successfully authenticated with the NTLM server) and **Unknown** (those who were not authenticated) types can be used in filtering policies.

NTLM authentication can work both with a proxy explicitly set in the user's browser (this is the standard mode) and in the transparent mode with no proxy set in the browser. DCFW is configured identically regardless of the authorization mode.

To configure authorization using an NTLM server, follow these steps:

Name	Description
Step 1. Configure time synchronization with the domain controller.	In DCFW settings, enable time synchronization with NTP servers. Specify the IP addresses of the domain controllers as the primary and (optionally) secondary NTP server.
Step 2. Create a DNS record for DCFW.	On the domain controller, create DNS records that correspond to DCFW and to be used as the auth.captive and logout.captive domains (e.g., auth.domain.loc and logout.domain.loc). Specify the address of a DCFW interface connected to the **Trusted** network as an IP address.
Step 3. Change the **Captive portal auth domain** address.	In the **General settings** section, change the **Captive portal auth domain** and (optionally) **Captive portal logout domain** addresses. For the **Captive portal auth** domain, specify the DNS record created at the previous step. For the **Captive portal logout** domain, specify the DNS record created at the previous step. For more details on changing the addresses of the captive portal's Auth and Logout domains, see the Captive Portal Configuration section.
Step 4. Add an NTLM server.	In the **Auth servers** section, click **Add**, select **Add NTLM server**, and specify the display name for the server and Windows domain name. For NTLM authentication to work correctly, the domain name specified here must resolve into the IP addresses of the domain controllers.
Step 5. Create a captive portal rule with NTLM authentication.	Configure the captive portal for using the NTLM authentication method. The captive portal is described in more detail in the following chapters.
Step 6. Enable HTTP(S) service access for the zone.	In the **Zones** section, enable access to the HTTP(S) proxy service for the zone to which the users who are authorized using NTLM are connected.
Step 7. For standard-mode authorization, configure the proxy on the user computers.	On user computers, turn on mandatory proxy use and specify the IP address of a trusted interface of DCFW as the proxy address. **Important!** You can use a domain name instead of an IP address, but the important thing for NTLM is that this name

Name	Description
	should not come from the Active Directory domain, otherwise the Windows computer will try to use Kerberos authentication. **Important!** The names used as the auth.captive and logout.captive domain in DCFW settings should not come from the Active Directory domain, otherwise a Windows-based computer will attempt to use Kerberos authentication.
Step 8. For transparent-mode authorization, configure automatic browser-based user authentication for all zones.	On user computers, go to **Control panel → Internet options → Security**, select the **Internet** zone **→ Custom level → User Authentication → Logon** and choose **Automatic logon with current name and password**. Repeat this setting for all other zones configured on this computer (Local intranet, Trusted sites).

Kerberos Authentication Method

The Kerberos option enables transparent (i.e., without requesting a username and password) authorization of Active Directory domain users. When authorizing via Kerberos, DCFW works with domain controllers that authenticate users for Internet access.

Kerberos authentication can work both with a proxy explicitly set in the user's browser (this is the standard mode) and in the transparent mode with no proxy set in the browser.

To configure authorization using Kerberos, follow these steps:

Name	Description
Step 1. Create DNS records for DCFW.	On the domain controller, create DNS records that correspond to DCFW and to be used as the auth.captive and logout.captive domains (e.g., auth.domain.loc and logout.domain.loc). Specify the address of a DCFW interface connected to the **Trusted** network as an IP address. **Important!** For correct operation, create type A records rather than CNAME.

Name	Description
Step 2. Create a user for DCFW.	Create a user in the AD domain, such as kerb@domain.loc, with the **password never expires** option. Set a password for user kerb. **Important!** Do not use characters from national alphabets, such as Cyrillic, in the names of the kerb user or in the Active Directory organization units where you plan to create this user account. **Important!** Do not use the user created for the LDAP connector as the kerb user. A separate user account needs to be used.
Step 3. Create a keytab file.	On the domain controller, create a keytab file by invoking the following command **as an administrator** (in one line!): ktpass.exe /princ HTTP/auth.domain.loc@DOMAIN.LOC /mapuser kerb@DOMAIN.LOC /crypto ALL /ptype KRB5_NT_PRINCIPAL /pass * /out C:\utm.keytab Enter the password for user kerb. **Important!** The command is case-sensitive. In the above example: auth.domain.loc is the DNS record created for the UserGate server at Step 1; DOMAIN.LOC is the Kerberos realm domain (UPPERCASE required!); and kerb@DOMAIN.LOC is the username in the domain created at Step 2 (again, UPPERCASE required for the realm domain name!).
Step 4. Configure DNS servers in UserGate.	In the UserGate settings, set the domain controller's IP addresses as the system DNS servers.
Step 5. Configure time synchronization with the domain controller.	In UserGate settings, enable time synchronization with NTP servers. Specify the IP addresses of the domain controllers as the primary and (optionally) secondary NTP server.
Step 6. Change the **Captive portal auth domain** address.	In the **General settings** section, change the **Captive portal auth domain** and (optionally) **Captive portal logout domain** addresses to the DNS records created at the previous step. For more details on changing domain addresses, see the section G eneral Settings.
Step 7. Create an LDAP connector and upload the keytab file to it.	Create an authentication server of type LDAP connector and upload the keytab file obtained at the previous step. **Important!** Do not use the special Kerberos user created earlier as the user for the LDAP connector. A separate user account needs to be used. For more details on configuring an LDAP connector, see the section LDAP Connector.

Name	Description
Step 8. Create a captive portal rule with Kerberos authentication.	Configure the captive portal for using the Kerberos authentication method. For more details on the captive portal, see the Captive Portal Configuration section.
Step 9. Enable HTTP(S) service access for the zone.	In the **Zones** section, enable access to the HTTP(S) proxy service for the zone to which the users authorized using Kerberos are connected.
Step 10. For standard-mode authorization, configure the proxy on the user computers.	On user computers, turn on mandatory proxy use and specify the proxy as the UserGate FQDN created at Step 3.
Step 11. For transparent-mode authorization, configure automatic browser-based user authentication for all zones.	On user computers, go to **Control panel → Internet options → Security**, select the **Internet** zone **→ Custom level → User Authentication → Logon** and choose **Automatic logon with current name and password**. Repeat this setting for all other zones configured on this computer (Local intranet, Trusted sites).

HTTP Basic Authentication Method

The Basic option enables authorization of users with an explicitly set proxy using a local and LDAP user database. This authentication type is not recommended for use because it transmits the username and password over the network in plain text. The HTTP Basic authentication can be used to automatically authorize command-line utilities that need Internet access, for example:

curl -x 192.168.179.10:8090 -U user: password http://www.msn.com

To configure HTTP Basic authorization, follow these steps:

Name	Description
Step 1. Create a DNS record for DCFW.	On the domain controller, create DNS records that correspond to DCFW and to be used as the auth.captive and logout.captive domains (e.g., auth.domain.loc and logout.domain.loc). Specify the address of a DCFW interface connected to the **Trusted** network as an IP address.
Step 2. Change the **Captive portal auth domain** address.	In the **General settings** section, change the **Captive portal auth domain** and (optionally) **Captive portal logout domain** addresses. For the **Captive portal auth** domain, specify the DNS record created at the previous step.

Name	Description
	For the **Captive portal logout** domain, specify the DNS record created at the previous step. For more details on changing the addresses of the captive portal's Auth and Logout domains, see the Captive Portal Configuration section.
Step 3. Create a captive portal rule with HTTP Basic authentication.	Configure the captive portal for using the HTTP Basic authentication method. In addition to configuring the **HTTP Basic** method itself, you also need to add the user database that will be used for authentication (e.g., add the Local user or LDAP server authentication methods). The captive portal is described in more detail in the following chapters.
Step 4. Enable HTTP(S) service access for the zone.	In the **Zones** section, enable access to the HTTP(S) proxy service for the zone to which the users authorized using HTTP Basic are connected.
Step 5. Configure a proxy on user computers.	On user computers, turn on mandatory proxy use and specify the IP address of a trusted interface of DCFW as the proxy address.

Authentication Profiles

An authentication profile can be used to specify a set of methods and settings for user authorization to be used later in various DCFW subsystems, such as captive portal, VPN, web portal, etc. To create an authentication profile, go to the **Users and Devices → Auth profiles** section, click **Add**, and specify the required parameters:

Name	Description
Name	Profile name.
Description	Profile description.
MFA profile	The multi-factor authentication profile. This needs to be created in advance in the **MFA profiles** section if multi-factor authentication is to be used. The profile defines the method of one-time password delivery for the second authentication factor. For more details on configuring an MFA profile, see later in the corresponding chapter. **Important!** Multi-factor authentication is only possible with authentication methods that allow the user to enter a one-time

Name	Description
	password, i.e., where the user explicitly enters their credentials in the auth page's web form. Therefore, multi-factor authentication cannot be used with Kerberos or NTLM.
Idle time	This parameter determines the time in seconds after which DCFW will re-classify a user from type **Known** to type **Unknown** when there is no activity from the user (which means no network packets coming from the user's IP address).
Expiration time	This parameter determines the time in seconds after which DCFW will re-classify a user from type **Known** to type **Unknown**. When this time elapses, the user will have to re-authenticate at the captive portal.
Maximum auth failures (local users)	The allowed number of failed authentication attempts via the captive portal after which the user account is locked.
Local user lockout time	The time for which the user account is locked on reaching the specified number of failed authentication attempts.
Authentication methods	The user authentication methods added earlier, for example, an Active Directory or RADIUS server. If there are multiple authentication methods, they will be used in the order they are listed in the console. Built-in authentication mechanisms can also be used, such as: • **Local user** authentication: authentication using a local user database. • **Policy accept**: authentication is not required, but before the user is granted access to the Internet, they must consent to the network usage policy. This authentication type must be used in conjunction with a captive portal profile that uses a captive portal policy auth page. • **HTTP Basic**: authentication using the legacy HTTP Basic method. • **Kerberos authentication**: authentication using the Kerberos protocol.

Captive Portal Configuration

The captive portal makes it possible to authorize **Unknown users** with the help of authorization methods that use Active Directory, RADIUS, TACACS+, SAML IDP,

Kerberos or NTLM directories or a local user database. Moreover, using the captive portal, you can configure user self-registration with email or SMS verification.

Remember that:

- Identified users, such as those with an explicitly set IP address in the user profile or those identified using authorization agents for terminal servers or Windows systems, are not authorized at the captive portal. These users are already classified as **Known** and do not require further identification.
- Captive portal authorization is only possible for HTTP and HTTPS protocols. For example, if you have created a firewall rule that allows Internet access using the FTP protocol only for **Known users**, users will not get Internet access using this protocol until they are identified; that is, they launch a browser on their device and pass authorization at the captive portal.
- To authorize users that use HTTPS, you need to configure SSL inspection, or authorization will not work.
- If the captive portal uses the Active Directory authorization method, the user must specify their login name as DOMAIN\username or username@domain.

To configure the captive portal, follow these steps:

Name	Description
Step 1. Create an authorization method, e.g., Active Directory domain-based authorization.	In the DCFW console, go to the **Users and devices → Auth servers** section, click **Add**, and create an authentication server.
Step 2. Create an authentication profile with the desired authorization methods.	In the DCFW console, go to the **Users and devices → Auth profiles** section, click **Add**, and create an authorization profile using the previously created authorization method.
Step 3. Create a captive profile with the desired authentication profile.	In the DCFW console, go to the **Users and devices → Captive profiles** section, click **Add**, and create a captive profile using the previously created authorization profile.
Step 4. Create a captive portal rule.	A captive portal rule determines the type of traffic to which the user authentication methods specified in the captive profile should be applied. In the DCFW console, go to the **Users and devices → Captive portal** section, click **Add**, and create a captive portal rule.

Name	Description
Step 5. Configure DNS for the auth.captive and logout.captive domains.	The internal **auth.captive** and **logout.captive** domain names are used by DCFW for user authorization. If the clients use DCFW as a DNS server, you do not need to do anything. Otherwise, you need to specify the IP address of the DCFW interface connected to the client network as the IP address for these domains. An alternative solution is to configure the **Captive portal auth domain** and **Captive portal logout domain** settings. For more details on these settings, see the section General Settings.

You can find an in-depth discussion of how to add authorization methods in the previous chapters. Let us now consider the creation of a captive profile and captive portal rules in more detail.

To create a captive profile, go to the **Captive profiles** section, click **Add**, and provide the desired settings:

Name	Description
Name	Captive profile name.
Description	Captive profile description.
Auth page template	Select a template for the auth page. You can create auth page templates in the **Libraries → Response pages** section. If you need to configure user self-registration with SMS or email verification, select the corresponding template type (Captive portal: SMS auth/ Captive portal: Email auth).
Authentication mode	The method that DCFW will use to remember this user. There are two options: • **Use IP address**. Having successfully authorized the user at the captive portal, DCFW saves their IP address, and all subsequent connections from that IP address will be associated with this user. This method provides identification of data transmitted over any TCP/IP family protocol, but will not work correctly if there is a NAT-connection between users and DCFW. This is the recommended value set by default. • **Use cookie**. After a user successfully authenticates through the captive portal, DCFW adds a cookie to the user's browser to identify their subsequent connections. This method allows authorization of users who are behind a NAT device but only for the HTTP(S) protocol and only in the same browser that was used for Captive portal authorization. Moreover, to authorize the user's HTTPS sessions, DCFW will decrypt all HTTPS

Name	Description
	connections on a mandatory basis. For firewall rules, a user authenticated using a cookie will always be classified as **Unknown**.
Auth profile	The authorization profile created earlier that defines the authentication methods to use.
Authentication mode	It is possible to authenticate using login and password via RADIUS s erver (AAA) or certificates (PKI).
User certificate profile	When PKI-based authentication is used, specify a pre-configured user certificate profile here.
Redirect URL	URL to redirect the user to after successful authentication using the Captive portal. If not specified, the user is redirected to the URL they requested.
Allow browsers to keep auth	Enables storing of the authorization in the browser for the specified time in hours. To store the authorization information, cookies are used.
Show AD/LDAP domain selector on Captive portal auth page	If enabled, this parameter allows the user to select the domain name from a list on the auth page if the Active Directory authentication method is used. If this parameter is not enabled, the user must explicitly specify the domain as DOMAIN\username or username@domain.
Protect with CAPTCHA	If this option is enabled, the user will be prompted to enter a code shown to them on the captive portal's auth page. This is recommended to protect against bots that guess user passwords.

To set up user self-registration with password verification using SMS or email, you need to configure settings on the **Guest users registration** tab. Remember to use the appropriate template type in this case (Captive portal: SMS auth/ Captive portal: Email auth).

Name	Description
Notification profile	The notification profile that will be used for sending information on the newly created user and their password. Two types of notification are possible, SMS and email. For more details on creating a notification profile, see the Notification Profiles chapter.
From	The person or entity in whose name notifications will be sent.

Name	Description
Notification subject	The subject of notifications (only for email notifications).
Notification body	The body of the notification message. In the message body, you can use special variables named {login} and {password} that will be replaced with the username and password, respectively.
Expiration date and time	The date and time when the guest account will be disabled.
Guest user TTL	The length of time from the guest user's first login after which their user account will be disabled.
Password length	Sets the password length for a guest user.
Password complexity	Sets the password complexity for a guest user. The available options are: • Numeric • Alphanumeric • Alphanumeric+special.
Groups	The groups to which the created guest users will be added.

To create a captive portal rule, go to the **Captive portal** section, click **Add**, and provide the desired settings:

Name	Description
Name	The name of the captive portal rule.
Description	A description of the captive portal rule.
Captive profile	Select a captive profile created earlier. An option is available called **Skip captive portal page** which, if enabled, waives the authentication requirement.
Enable logging	If this is enabled, instances of the rule being triggered will be recorded in the corresponding statistics log.
Source	The source addresses. You can use a specific zone, such as the LAN zone, or an IP address range as the source. Country IP addresses (GeoIP) can also be used. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.

Name	Description
	Important! The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Destination	The destination addresses. You can use a specific zone, such as the WAN zone, or an IP address range as the destination. Country IP addresses (GeoIP) can also be used. **Important!** The maximum number of GeoIPs that can be specified is limited to 15. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Categories	The URL filtering categories to which the rule will be applied. You need to have the appropriate license for URL filtering.
URL	The URL lists to which the rule will be applied.
Time	The time when this rule will be active.
Usage	The trigger statistics for the rule: the total trigger count and the time of the first and last trigger. To reset the trigger count, select the rules in the list and click **Reset hit counts**.
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

By creating several captive portal rules, you can configure different user identification policies for different zones, URL categories, and time.

Note

The conditions specified in the rule's tabs are combined with a Boolean AND, i.e., all conditions must be met to trigger the rule. If you need to use the OR logic instead, this can be achieved by creating several rules.

> **Note**
> **The rules are applied in the order they are listed in the console. You can reorder the rules using the corresponding buttons.**

> **Note**
> **When there are multiple matching rules, only the first triggered rule is applied.**

To change the user after logging in to the system or to log out, go to http://logout.captive and click **Logout**.

Terminal Server Users

A terminal server is used to provide remote access to a desktop or console for users. Generally, one terminal server provides service to multiple users, sometimes even dozens or hundreds of users. Identifying terminal server users is a problem because all server users have the same IP address, and DCFW cannot correctly identify the network connections of the individual users. As a solution to this problem, use of a dedicated terminal server agent is offered. Each user is allocated a port range that is used for their connection, i.e., the original ports are substituted with the ports from the range allocated to the user.

The terminal server agent must be installed on all terminal servers that need user identification. The agent is a service that transmits information to UserGate DCFW about the users of the terminal server and their network connections. Due to the way TCP/IP works, a terminal server agent can only identify user traffic that utilizes the TCP and UDP protocols. Protocols other than TCP/UDP, such as ICMP, do not allow identification.

For correct user identification when Active Directory authorization is used on the terminal servers, an active Active Directory connector server is required.

To start using terminal server user authentication, follow these steps:

Name	Description
Step 1. Allow the **Authentication agent** service in the desired zone.	In the **Network → Zones** section, allow the **Authentication agent** service for the zone on the terminal servers' side.

Name	Description
Step 2. Set a password for terminal server agents.	In the DCFW console, go to the **UserGate → Settings → Modules** section, click the **Configure** button next to the **Password for terminal server agent** entry, and set a password for terminal server agents.
Step 3. Install the terminal server agent.	Install the terminal server agent on all servers that require user identification. During the installation, specify the DCFW IP address and the password set at the previous step.
Step 4. Add the desired servers in the DCFW console.	In the **Users and devices → Terminal servers** section, add the terminal server agents, specifying the host name and address. After receiving the data from the host specified in the settings, provided that the password set at Step 2 is correct, user authentication will be enabled automatically. Upon a DCFW version update, the terminal server agents that were displayed in the web console earlier will continue working.

UserGate will now receive user information.

The terminal server agent enables not only domain users to be authenticated but also local users of a terminal server by adding the following parameter to its configuration file (%ALLUSERSPROFILE%\Entensys\Terminal Server Agent\tsagent.cfg):

LocalDomain = 1

When this parameter is on, the user information will be transferred to DCFW in the "server_username" for local users or "domain_username" for domain users format.

In addition, these users will need to be added to DCFW as local users. For details on adding users, see the Users section. When adding, you must specify the **Login** in the format specified above; password is not required.

This option is intended primarily for terminal servers that are not included in a domain and are logged in to using local credentials.

After editing the configuration file, make sure to restart the terminal agent service.

Note

Only letters, numbers, and the underscore character are allowed in the computer name; hyphens are prohibited.

You can change the settings of a terminal server by editing the configuration file of the terminal server authorization agent. After making the changes, make sure to restart the authentication agent.

The settings that can be configured in the tsagent.cfg file are listed below:

- **TimerUpdate**: the time interval in seconds between updates.
- **MaxLogSize**: the maximum size of the service log in MB.
- **SharedKey**: the password for connecting the agent.
- **SystemAccounts**: can take values of **0** or **1**. **SystemAccounts=1** enables transmission of information on system accounts connections (system, local service, network service) and the connection ports they use to DCFW.
- **FQDN**: can take values of **0** or **1**. **FQDN=1** indicates that a FQDN (Fully Qualified Domain Name) is used, e.g., "example.com" as opposed to "example".
- **ServerPort**: the DCFW port number that accepts the authorization agent's connection. By default, UDP port 1813 is used.
- **ServerAddress**: the IP address of the UserGate device that accepts the connection from the authorization agent.
- **UserCount**: the maximum number of users to create.
- **BlockDNS**: can take values of **0** or **1**. With **BlockDNS=1** the source port is substituted with a free port from the user-allocated port range when sending DNS requests (UDP:53); with **BlockDNS=0**, DNS traffic is sent without port substitution.
- **BlockUDP**: can take values of **0** or **1**. With **BlockUDP=1**, the source port is substituted with a free port from the user-allocated port range when sending UDP traffic; with **BlockUDP=0**, the traffic is sent without port substitution.
- **ExcludeIP**: if multiple IP addresses are configured on the terminal server, they will all be used for user authentication. The ExcludeIP parameter allows restriction of users' Internet access from certain IP addresses used by the terminal server.
    - IP addresses in the x.x.x.x format and/or subnet addresses in the x.x.x.x/n format are specified separated by semicolons (for example, **ExcludeIP=x.x.x.x/n; x.x.x.x**).
    - Spaces are allowed between addresses in the list, they are ignored (for example, **ExcludeIP=x.x.x.x/n; x.x.x.x;y.y.y.y**).

    - If there are spelling errors in the addresses in the line, they will be reflected in the logs when the agent starts. Only correctly specified addresses will be used. The number of used addresses from the list is written to the log when the agent starts.

    - If, as a result of filtering, all addresses are excluded from the distribution, then a log entry is made (once) in the form: **GetIPAddressList: IP list is blocked by ExceptIP**. If a non-empty distribution is later generated, a log entry is made in the form: **GetIPAddressList: IP list is not blocked by ExceptIP anymore**.

- **ExcludePorts**: the range of ports to be excluded from being substituted with ports from the user-allocated port range. Specified as: **ExcludePorts=port1-port2**.

- **NAT_IP**: required when there is a NAT between the terminal server and UserGate. The terminal server's IP address is substituted with an address from the specified range. The IP addresses are specified as: **NAT_IP="12.3.4-1.1.1.1;2.2.2.2-5.5.5.5"**.

To exclude certain addresses and/or subnets from distribution by the terminal agent, in addition to adding the **ExcludeIP** parameter to the tsagent.cfg configuration file, it can also be activated in the server registry as follows:

- Added as a string parameter to the Windows registry key [HKEY_CURRENT_USER\Software\Policies\Entensys\Auth Client]. In this case, the parameter settings will only apply to this user.

- Added as a string parameter to the Windows registry key [HKEY_LOCAL_MACHINE\Software\Policies\Entensys\Auth Client]. In this case, the parameter settings will apply to all users of this system.

The order of searching for ExcludeIP parameter settings in the system is as follows: first, the parameter is searched in the registry key [HKEY_LOCAL_MACHINE\Software\Policies\Entensys\Auth Client], then in the registry key [HKEY_CURRENT_USER\Software\Policies\Entensys\Auth Client], then in the tsagent.cfg file.

MFA (Multi-Factor Authentication) Profiles

Multifactor authentication is an identification and authentication mode where two or more different types of authentication data (factors) are used. This additional level

of security provides more effective protection from unauthorized access to the account.

DCFW supports multi-factor authentication using a username and password as the first authentication factor, and the following types as the second factor:

- **TOTP** (Time-based One Time Password) token: a TOTP token creates a time-based single-use password, i.e., time is a parameter here. For more details on TOTP, see https://en.wikipedia.org/wiki/Time-based_One-time_Password_Algorithm. The token may come in the form of various devices or software installed on users' smartphones, such as Google Authenticator.
- **SMS**: a one-time password sent by SMS. To receive SMS messages, each user must have their phone number provided in their local DCFW user account or Active Directory domain user account.
- **Email**: a one-time password sent by email. To receive emails, each user must have their email address entered in their local DCFW user account or Active Directory domain user account.

To configure multi-factor authentication, follow these steps:

Name	Description
Step 1. Configure captive-portal authorization.	Multi-factor authorization works only when users are authorized using the captive portal. For more details, see the relevant section.
Step 2. Create a multi-factor authorization profile.	In the **Users and devices → MFA profiles** section of the console, create a multifactor authorization profile with the desired second-factor delivery settings. Three delivery types are available: • **MFA by TOTP**: deliver the second authorization factor using TOTP tokens • **MFA by SMS**: deliver the second authorization factor using SMS • **MFA by email**: deliver the second authorization factor using email.

For **MFA by TOTP**, provide these settings:

Name	Description
Name	The name of the MFA profile.
Description	A description of the MFA profile.

Name	Description
TOTP initialization	To receive TOTP tokens, you need to initialize the client device or software by entering a unique key into the device. The TOTP initialization code can be communicated by: • Showing it on the captive portal page after first successful login. To do this, select **Show key on captive portal page**. • Sending it by SMS. To receive SMS messages, each user must have their phone number provided in their local DCFW user account or Active Directory domain user account. This option requires selecting an appropriate SMS sending profile (SMPP profile) created earlier. • Sending it by email. To receive emails, each user must have their email address entered in their local DCFW user account or Active Directory domain user account. This option requires selecting an appropriate email sending profile (SMTP profile) created earlier.
Show QR code	Show a QR code on the captive portal page or in the email to facilitate TOTP device or software configuration.

If the user has lost the token, the administrator can trigger a mandatory re-initialization of the TOTP token by selecting this user in the user list (**Users and devices → Users**) and choosing the **Reset TOTP key** option. On the next login attempt, the user will be asked to re-initialize their token.

For **MFA by SMS**, provide these settings:

Name	Description
Name	The name of the MFA profile.
Description	A description of the MFA profile.
Auth delivery profile	The SMPP profile that will be used to send passwords by SMS. For more details on configuring profiles for sending SMS messages, see the Notification Profiles section.
From	The person or entity in whose name notifications will be sent.
Body	The body of the notification message. In the message body, you can use a special variable named {2fa_auth_code} that will be replaced by the one-time password.
MFA code lifetime	The validity period of the one-time password.

For **MFA by email**, provide these settings:

Name	Description
Name	The name of the MFA profile.
Description	A description of the MFA profile.
Auth delivery profile	The SMTP profile that will be used to send passwords by email. For more details on configuring profiles for sending email messages, see the Notification Profiles section.
From	The person or entity in whose name notifications will be sent.
Subject	Notification subject.
Body	The body of the notification message. In the message body, you can use a special variable named {2fa_auth_code} that will be replaced by the one-time password.
MFA code lifetime	The validity period of the one-time password.

UserID Agent

UserID is a technology that enables transparent user authentication on UserGate devices. Data sources for unique user identification include security logs from domain controller operating systems and application and access server logs where users are already authenticated.

To create policies that include users and groups, the firewall must map IP addresses to the users assigned to these addresses and retrieve information about the groups to which they belong. UserID provides several methods for performing this mapping. For example, to obtain user information, UserID can scan server logs for messages from authentication services. Users whose names cannot be mapped to IP addresses can be redirected to a special portal (Captive Portal) for authentication. To obtain group information, the firewall connects directly to LDAP servers.

Currently, UserID uses Microsoft Active Directory logs, syslog data, or RADIUS accounting messages (starting with software version 7.2.0) as data sources for authentication.

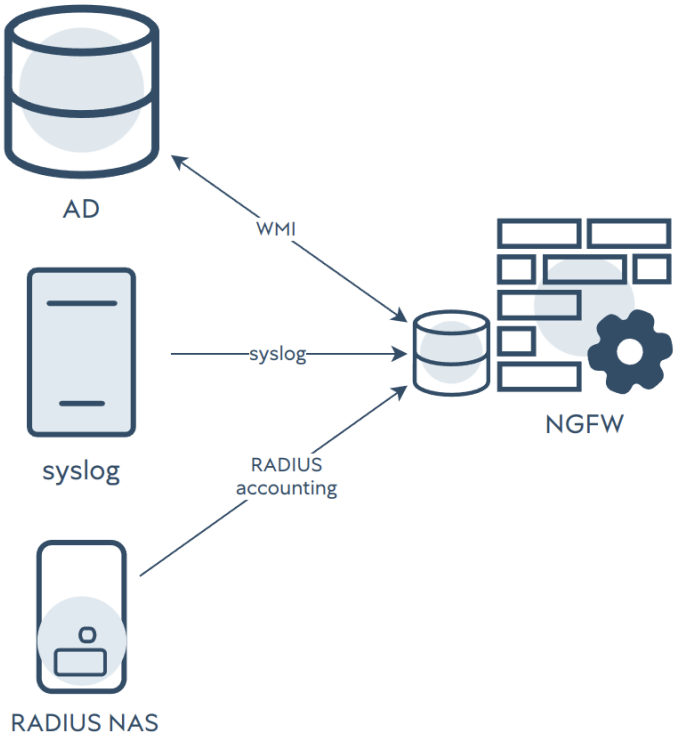
The operation of UserID is completely transparent to end users. This means that users do not need to authenticate on DCFW explicitly.

How UserID Works

Depending on the UserID usage scenario and configuration, the agent receives data on user authentication events using one of the following methods:

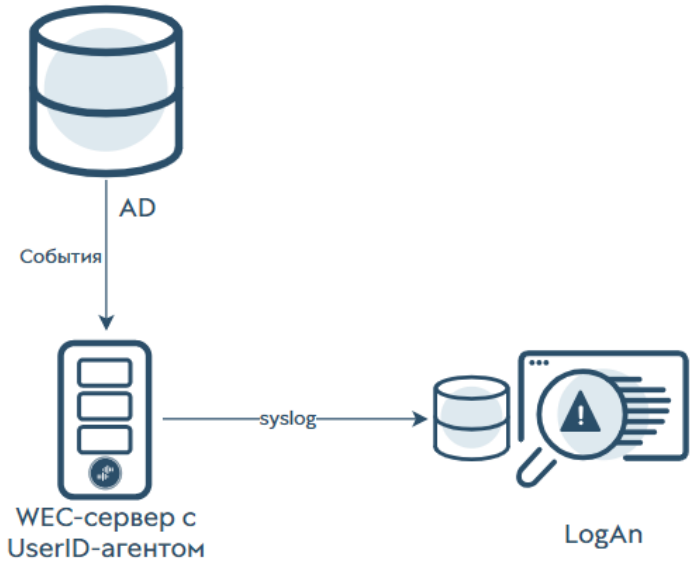
1. The node running the UserID agent collects data directly from authentication data sources by using configured connectors:

- The UserID agent can connect to the AD domain controller by means of the WMI technology to read security event logs.
- The UserID agent can receive messages from third-party servers using the syslog standard.
- The UserID agent can receive RADIUS accounting messages from third-party RADIUS NAS servers.



2. Working with a data source through an intermediary, which is a special software agent installed on a domain controller or event collector server (WEC):

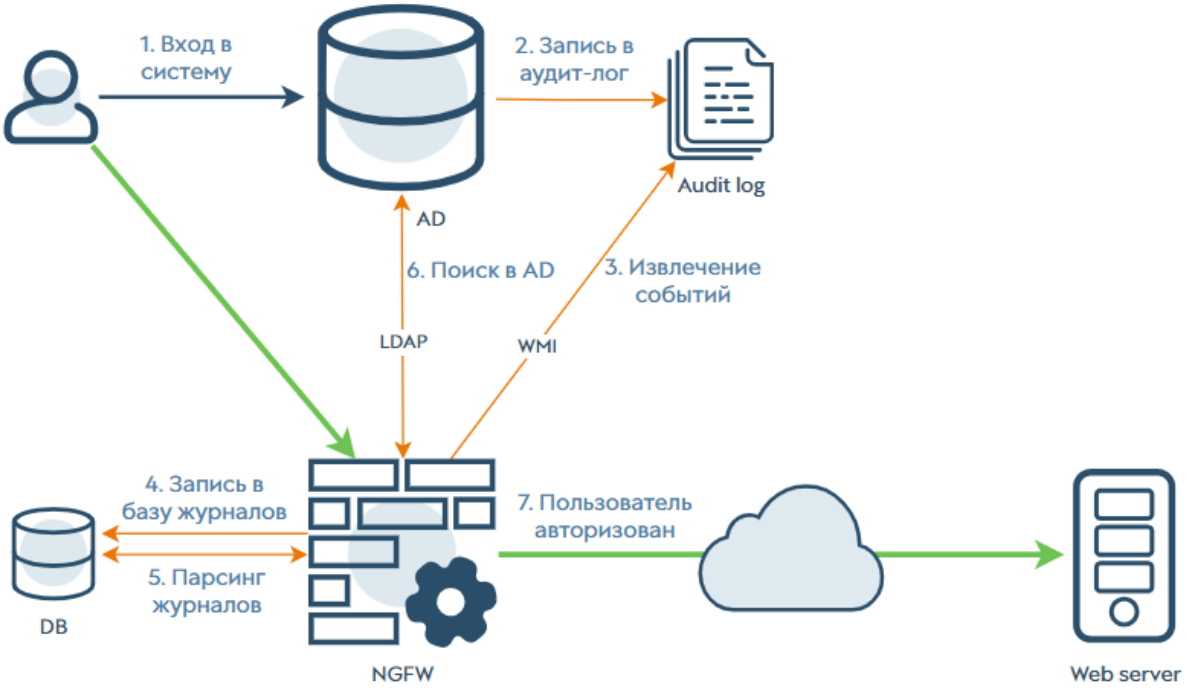
- The UserID software agent for AD/WEC is installed on a domain controller (AD) or a domain event collector server (WEC) to read the information necessary for user identification from Windows security logs and then forward it in syslog format to the UserID collector on LogAn (please see more details about the agent in the UserID agent for AD/WEC article).



The main advantages of this method of obtaining data from an AD domain are:

- There is no need to provide external access to the domain controller to collect user authentication data, as is required for providing access using the WMI technology.
- There is no need to create a separate account with special privileges in the domain for the nodes running the UserID agent.

Let's look at how UserID works using the example of a scenario involving interaction with Active Directory as a data source for user authentication via WMI.



The AD domain controller has security event auditing enabled, which records events by configured categories in a dedicated audit log.

Once you created and configured the UserID agent and Microsoft Active Directory connector on DCFW, the UserID agent begins sending WMI requests regularly to the AD controller to extract the following events from the audit log by their IDs:

- 4624 — a successful login;
- 4768 — a request for a Kerberos authentication ticket (TGT);
- 4769 — a request for a Kerberos authentication ticket (TGS);
- 4770 — a Kerberos authentication ticket update (TGS);
- 4627 — group membership information.

These events allow the UserID agent to receive information on user registrations and their group memberships. This information is stored in DCFW's dedicated system database.

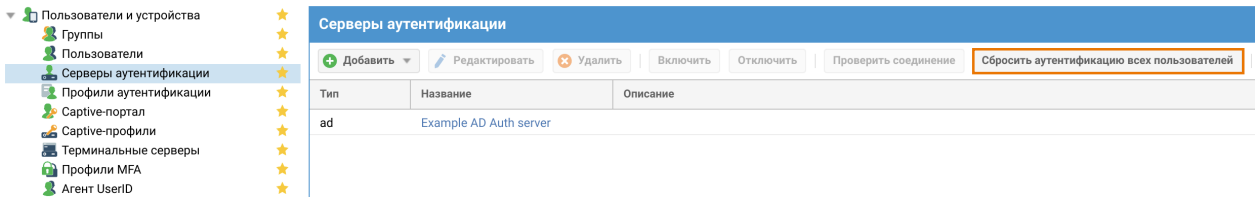
The UserID agent then accesses this database on a regular basis, extracting the username, domain, SID, IP address, and user group list information from records. This data is cached. The database search interval for the records can be configured in the UserID agent settings. The user data lifetime for the cache is configured in the UserID agent connector settings on DCFW.

If the user's group list is not retrieved, the UserID agent contacts the domain controller to obtain group information by means of LDAP in accordance with the configured authentication profile.

If conditions for certain users or groups are set up in the rules, then during network traffic processing DCFW accesses the cache to find information on IP addresses the users are registered with. This information is used to decide upon the packet handling method.

The user session termination can be enforced by the DCFW administrator. To do this, the administrator can reset all users or a specific user:

- In the admin web console, go to the **Settings → Users and devices → Auth servers** section, and click **Drop all users auth** button on the dashboard:



- In the CLI console, use the command:

```
Admin@nodename# execute termination user-sessions ip <IP-address>
```

In a scenario where syslog data source servers are used as user authentication data sources, the operating principle is similar, except that in this case DCFW acts as a syslog listener receiving messages from a syslog sender (the port number and protocol are set in the UserID agent settings with TCP port 514 used by default), and then filtering the required events out from the received data stream using configured filters from the "UserID agent syslog filters" library. In this case, username, IP address, and SID (optional) are saved in the database. To obtain information on the groups to which a user is registered, the UserID agent contacts the domain controller via LDAP in accordance with the configured UserID agent authentication profile.

In a scenario using RADIUS accounting messages as the source of user authentication data, the operating principle is generally similar. In this scenario, DCFW acts as a transit RADIUS server. It receives RADIUS accounting messages from NAS servers (via UDP port 1813), and verifies users on the AD domain controller via LDAP in accordance with the configured UserID agent authentication profile.

The UserID agent's configuration on DCFW is not cluster-based, meaning that it has to be performed on each node separately. Once configured, the UserID agent will operate and receive login and logout event data from the source logs independently.

In addition to DCFW, UserID can operate on LogAn (Log Analyzer) devices. For more information about using UserID on LogAn, see the Log Analyzer administrator's guide. Using LogAn allows you to scale UserID technology to other network devices. The principle of its operation on a LogAn device is similar to that for DCFW. Events found in the collected data are sent to other DCFWs according to the UserID Sharing policy and based on configured redistribution profiles. This policy allows to send different data to different DCFW nodes as necessary. While doing so, only the user's GUID, IP address, and the list of IDs of the groups the user belongs to are sent to DCFW. Such architecture allows you to use one or several LogAn servers to collect user information from various sources, and then selectively distribute this information to DCFW nodes by using a centralized approach.

UserID configuration algorithm

To configure UserID operation, you'll need to perform a number of steps on both the authentication data sources' side, and the DCFW side.

On the Data Source Side

When using Active Directory as a user authentication data source, you must enable security event auditing. The following categories are required:

- Audit LogOn;
- Audit LogOff;
- Audit Kerberos Authentication Service;
- Audit Group Membership;
- Audit Kerberos Service Ticket Operations.

When working with syslog data source servers, they must be configured to send logs to the UserID agent address (that is the DCFW IP address, with the port number and protocol specified in the UserID agent settings and TCP port 514 used by default).

When working with RADIUS NAS servers, they must be configured to send RADIUS accounting messages to the UserID agent address (i.e., the DCFW IP address, UDP port 1813).

On a DCFW side

You need to configure the following settings on a DCFW side:

- Create an authentication server for the UserID agent. For more information on creating and configuring an authentication server, see the Authentication servers section.
- Create UserID agent's authentication profile. For more details on authentication profile creation and configuration, please read the Authentication Profiles article.
- For the syslog data source servers scenario, activate UserID syslog collector service in access control settings for the zone where syslog sender will reside. For a scenario with RADIUS NAS servers, enable the "Authentication agent" service in the access control settings of the zones where the RADIUS NAS servers will be located. For more information on creating and configuring zones, see the Configuring Zones section.

- Create a UserID agent connector in accordance with an authentication data fetching method.
- Configure UserID agent's general settings.

Creating a UserID agent connector

To create an UserID agent connector in the DCFW's admin web console, go to the **Settings → Users and devices → UserID agent connectors** section. Click the **Add** button on the toolbar, and select the type of connector:

- Microsoft Active Directory;
- Syslog sender;
- RADIUS server.

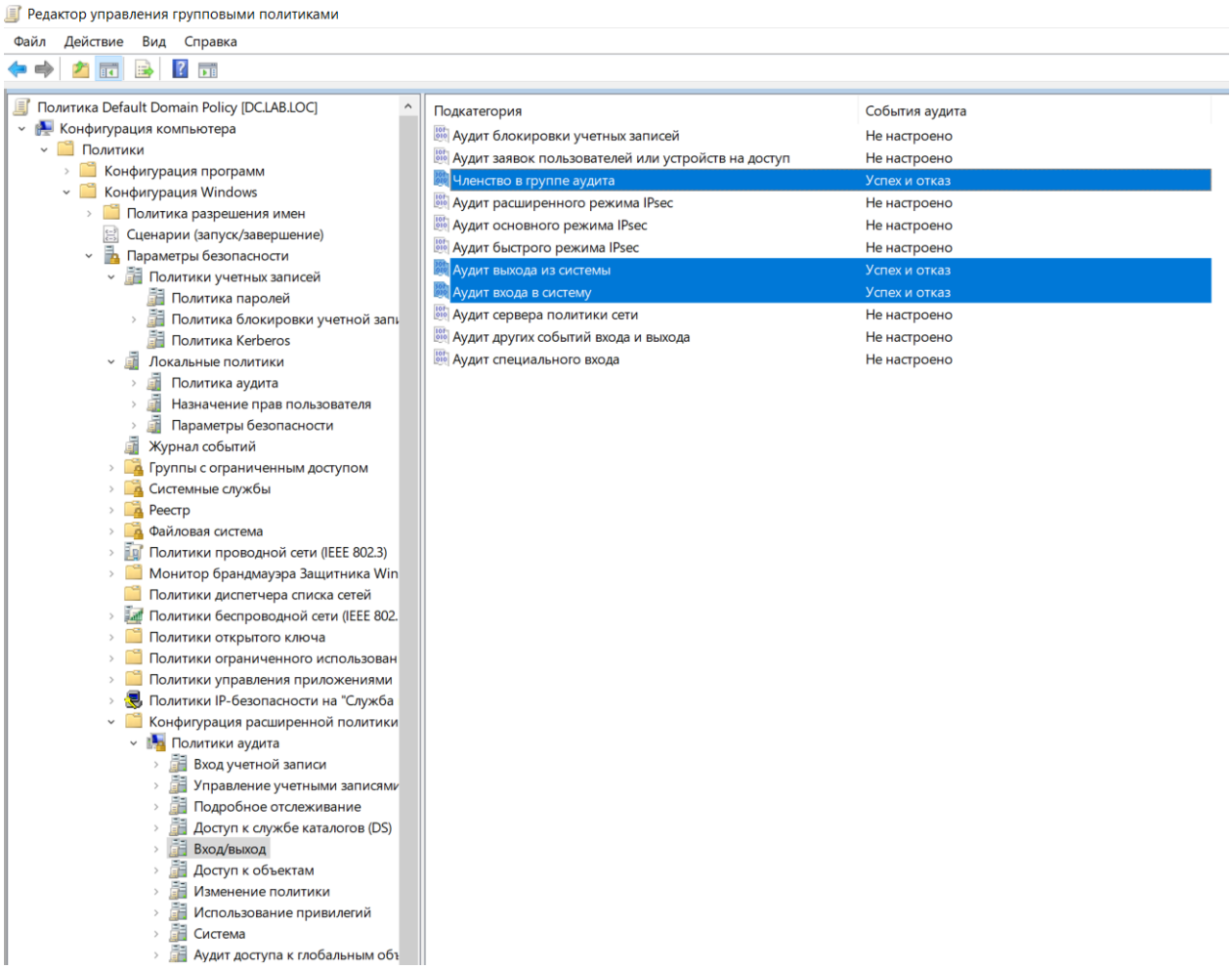
Microsoft Active Directory

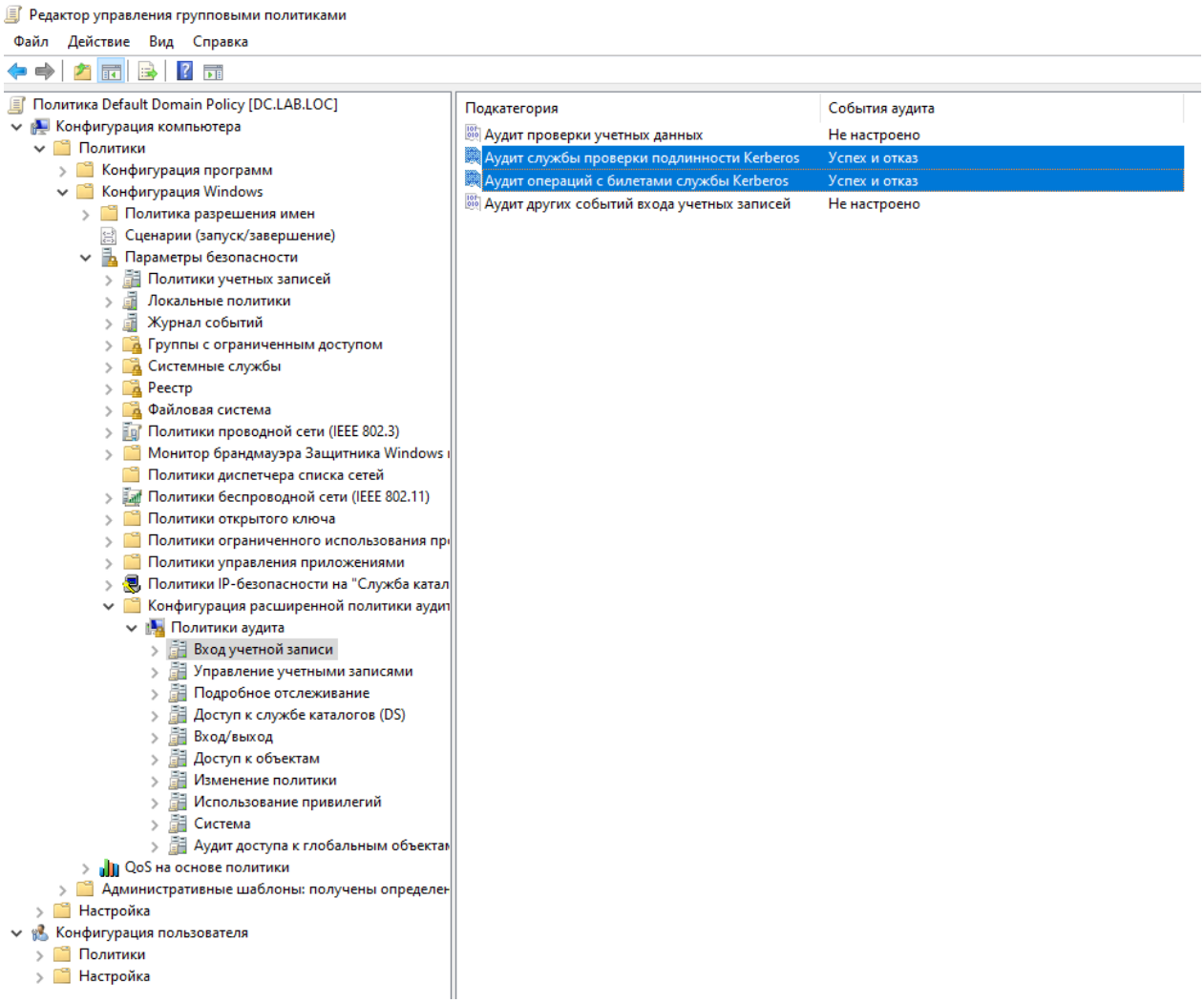
If Microsoft Active Directory is used as the source of information, you need:

1. Configure the event source.

2. Configure the UserID agent connector's settings for AD monitoring.

To enable event audit on the AD server, you need to edit **Audit policies** under the default **Domain policy** and **Extended policy configuration** using gpedit.msc, as indicated on the following screenshots:


Редактор управления групповыми политиками
Файл Действие Вид Справка
Политика Default Domain Policy [DC.LAB.LOC]
Конфигурация компьютера
Политики
Конфигурация программ
Конфигурация Windows
Политика разрешения имен
Сценарии (запуск/завершение)
Параметры безопасности
Политики учетных записей
Политика паролей
Политика блокировки учетной зап
Политика Kerberos
Локальные политики
Политика аудита
Назначение прав пользователя
Параметры безопасности
Журнал событий
Группы с ограниченным доступом
Системные службы
Реестр
Файловая система
Политики проводной сети (IEEE 802.3)
Монитор брандмауэра Защитника Win
Политики диспетчера списка сетей
Политики беспроводной сети (IEEE 802.
Политики открытого ключа
Политики ограниченного использован
Политики управления приложениями
Политики IP-безопасности на "Служба
Конфигурация расширенной политики
Политики аудита
Вход учетной записи
Управление учетными записями
Подробное отслеживание
Доступ к службе каталогов (DS)
Вход/выход
Доступ к объектам
Изменение политики
Использование привилегий
Система
Аудит доступа к глобальным об
Подкатегория
События аудита
Аудит блокировки учетных записей
Не настроено
Аудит заявок пользователей или устройств на доступ
Не настроено
Членство в группе аудита
Успех и отказ
Аудит расширенного режима IPsec
Не настроено
Аудит основного режима IPsec
Не настроено
Аудит быстрого режима IPsec
Не настроено
Аудит выхода из системы
Успех и отказ
Аудит входа в систему
Успех и отказ
Аудит сервера политики сети
Не настроено
Аудит других событий входа и выхода
Не настроено
Аудит специального входа
Не настроено



To execute WMI queries, you must create a user with the appropriate privileges using the procedure below.

> **Please note!**
>
> **These settings are required to enable the agent connection via WMI by using a restricted rights account.**

1. Create a user account on the domain controller:

   - Go to the **Start → Server manager → Tools → Active Directory — Users and computers** menu.
   - In the required organizational unit, create a **New user** for the UserID agent.

2. Configure group membership for the new user account:

   - Right-click the new UserID user account, and select **Properties**.

- Click the **Group membership** tab.
- Click **Add → Advanced → Search**.
- Select the following groups:
  - **DCOM users**
  - **Performance log users**
  - **Remote desktop users**
  - **Event log readers**
- Click **OK**.

3. Assign Distributed Component Object Model (DCOM) permissions:

- Go to the **Start → Administration → Component services** Windows menu. The **Component Services** window will open.
- Expand **Component services → Computers → My computer**.
- Right-click **My Computer** and select **Properties**. The **Properties: My Computer** window will open.
- Go to the **COM Security** tab.
- In the Permissions area, click **Change restrictions**.
- Make sure that **Local access** and **Remote access** are selected for **DCOM users**.
- Click **OK** to save the settings.
- In the **Properties: My Computer** window, under **Launch and activation permissions**, click **Change restrictions**.
- Make sure that **Local launch**, **Remote launch**, **Local activation** and **Remote activation** are selected for **DCOM users**.
- Click **OK** to save the settings, and then click **OK** one more time to close the **Properties: My computer** window.
- Select **File → Exit** to close the **Component services** window.

4. Configure WMI namespace security assignments:

- Go to the **Start → Run** menu.

- Type `wmimgmt.msc` and click **OK**.
- Right-click **WMI control (Local)** and select **Properties**.
- Open the **Security** tab.
- Click **Security → Add → Advanced → Search**.
- Select new user account, and click **OK** repeatedly until the **Security** window for Root opens.
- Click **Advanced**, and select the newly added user account.
- Click **Edit**.
- In the **Applied to:** menu, select **This namespace and subspace**.
- Make sure to select **Method execution**, **Enable account**, **Enable remotely**, and **Read security**.
- Click **OK**, until the wmimgmt window opens.
- Select **File → Exit** to close the wmimgmt window.

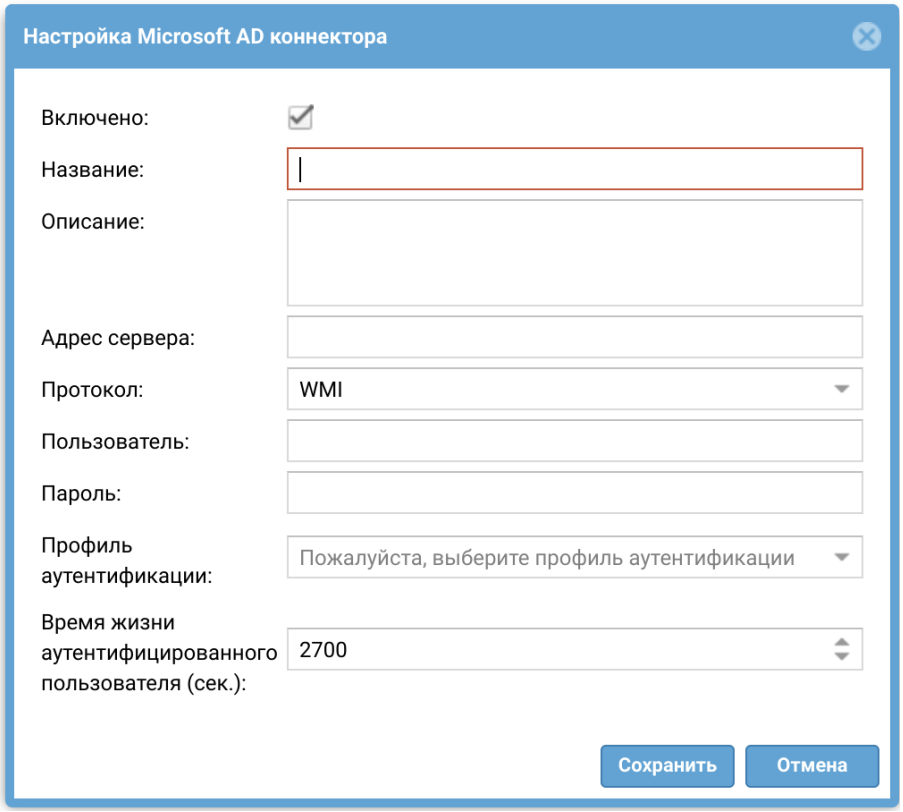
Please note!

The Windows KB5014692 update may cause the WMI access errors of: *NTSTATUS: NT_STATUS_ACCESS_DENIED* type. If this is the case, you can try to add the following information into the Windows registry:

```
Path : HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole\AppCompat

Value Name: "RequireIntegrityActivationAuthenticationLevel"

Type: dword

Value Data: 0x00000000
```

When using AD servers as event sources, the UserID agent performs WMI queries to search for successful login events (event ID 4624), Kerberos events (event numbers 4768, 4769, and 4770), and group membership events (event ID 4627).

In the DCFW's admin web console, go to the **Settings → Users and devices → UserID agent connectors** section, click **Add** on the dashboard, and select **Microsoft Active Directory** as the type of connector. Then specify the following information:



- **Enabled**: enable/disable receiving logs from the source.
- **Name**: source name.
- **Description**: an optional description of the source.
- **Server address**: Microsoft Active Directory server address.
- **Protocol**: AD access protocol (WMI).
- **User**: the username for connecting to AD.
- **Password**: the password for connecting to AD.
- **Authentication profile**: the name of the previously created authentication profile used to find users that are present in AD logs.
- **Expiration time (sec.)**: the period of time that, when expires, leads to user's session termination, i.e. deleting their information from the DCFW's cache. The default value is 2700 seconds (45 minutes).

Syslog

If the syslog sender is used as the information source, you must:

1. Configure the event source.

To ensure UserID syslog agent connector's correct operation, you need to configure the syslog data source server to send logs to the UserID agent's address. For more information, see the syslog sender documentation.

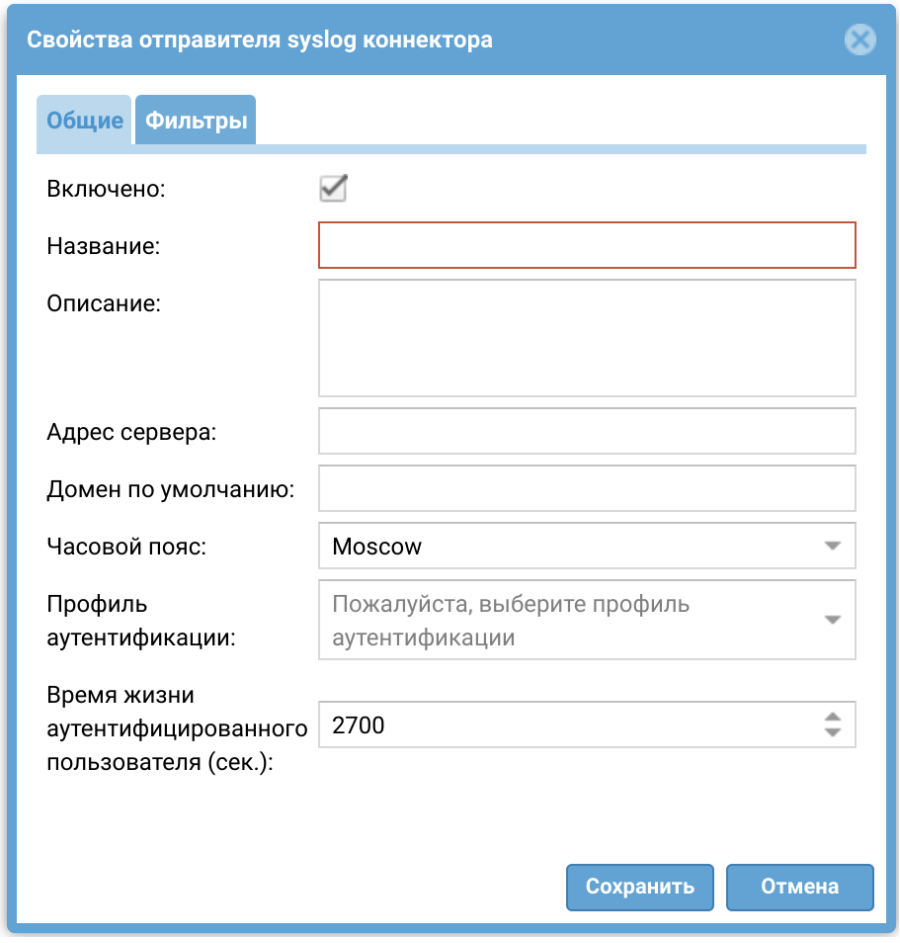
To view or edit the syslog server's parameters on DCFW, use the UserID agent's general settings.

2. Allow collecting information from remote devices using the syslog protocol.

In the access control settings for the zone in which the syslog sender is located, enable the "UserID syslog collector" service.

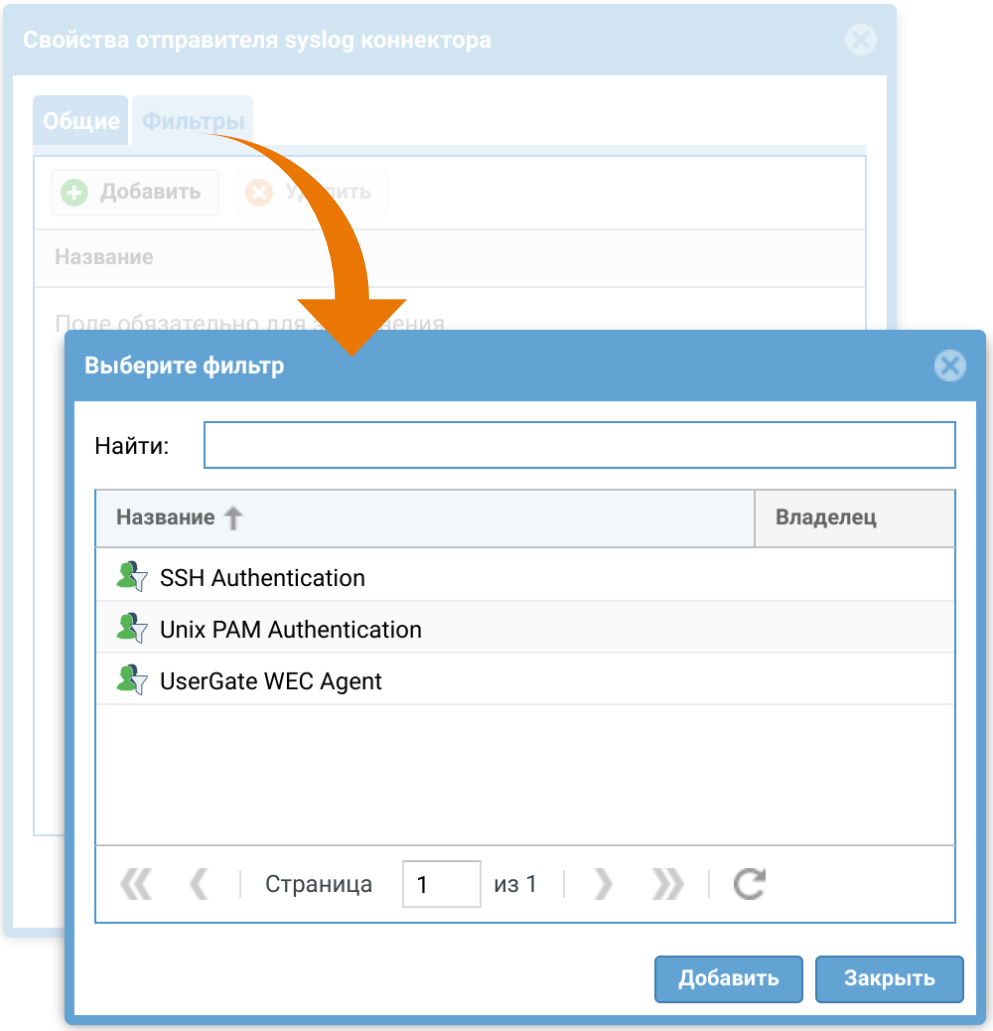
3. Configure the UserID agent connector's settings for syslog sender.

In the DCFW's admin web console, go to the **Settings → Users and devices → UserID agent connectors** section, click **Add** on the dashboard, and select **Syslog sender** as the type of connector. Then specify the following information:



- **Enabled**: enable/disable receiving logs from the source.
- **Name**: source name.
- **Description**: an optional description of the source.
- **Server address**: the host address from which DCFW will receive syslog events.
- **Default domain**: the name of a domain used to search for users found in syslog logs.
- **Timezone**: the time zone set on a source.
- **Authentication profile**: the authentication profile used to search for a user found in syslog logs.
- **Expiration time (sec.)**: the period of time that, when expires, leads to user's session termination, i.e. deleting their information from the DCFW's cache. The default value is 2700 seconds (45 minutes).

The **Filters** tab offers you the filters to find the necessary log entries.



To create and configure filters, use the **Libraries → UserID agent syslog filters** section. For more details, see UserID agent Syslog filters.

RADIUS accounting

(Available starting from the software version 7.2.0).

If RADIUS accounting messages are the source of information, you must:

1. Configure the event source.

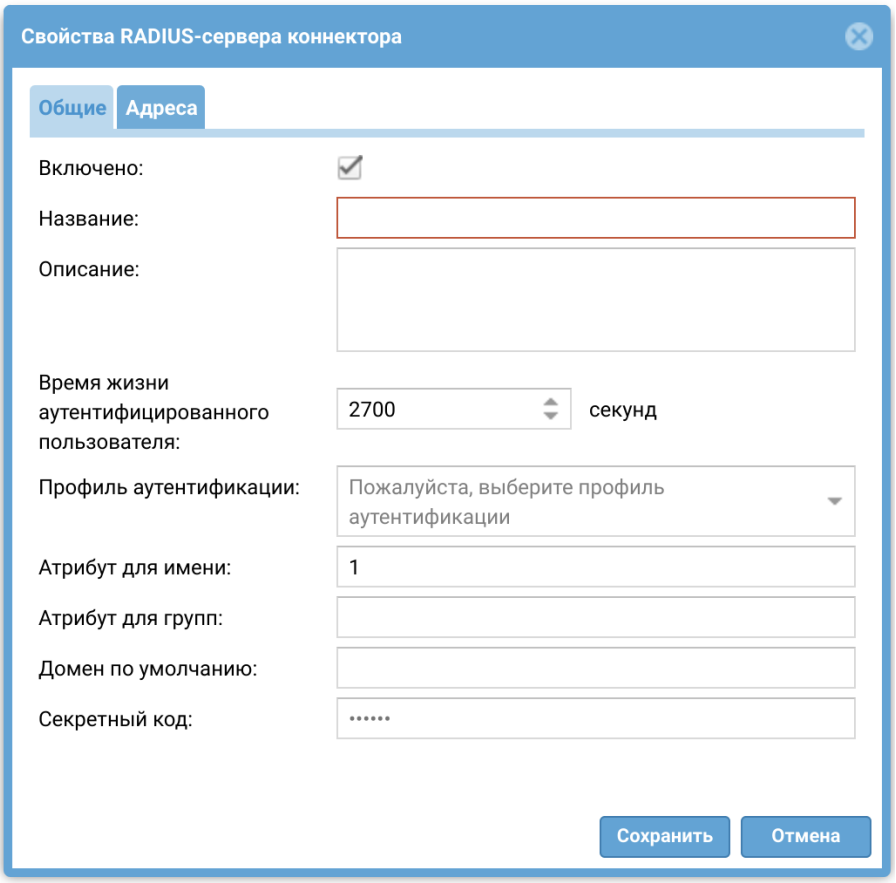
To ensure UserID agent connector's correct operation, you need to configure the NAS server to send RADIUS accounting messages to the UserID agent's address (UDP port 1813). For more information, see the NAS server documentation.

2. Allow receiving RADIUS accounting requests from remote devices.

In the access control settings for the zones where the NAS servers are located, enable the "Authentication agent" service.

3. Configure the UserID agent connector's settings for a RADIUS server.

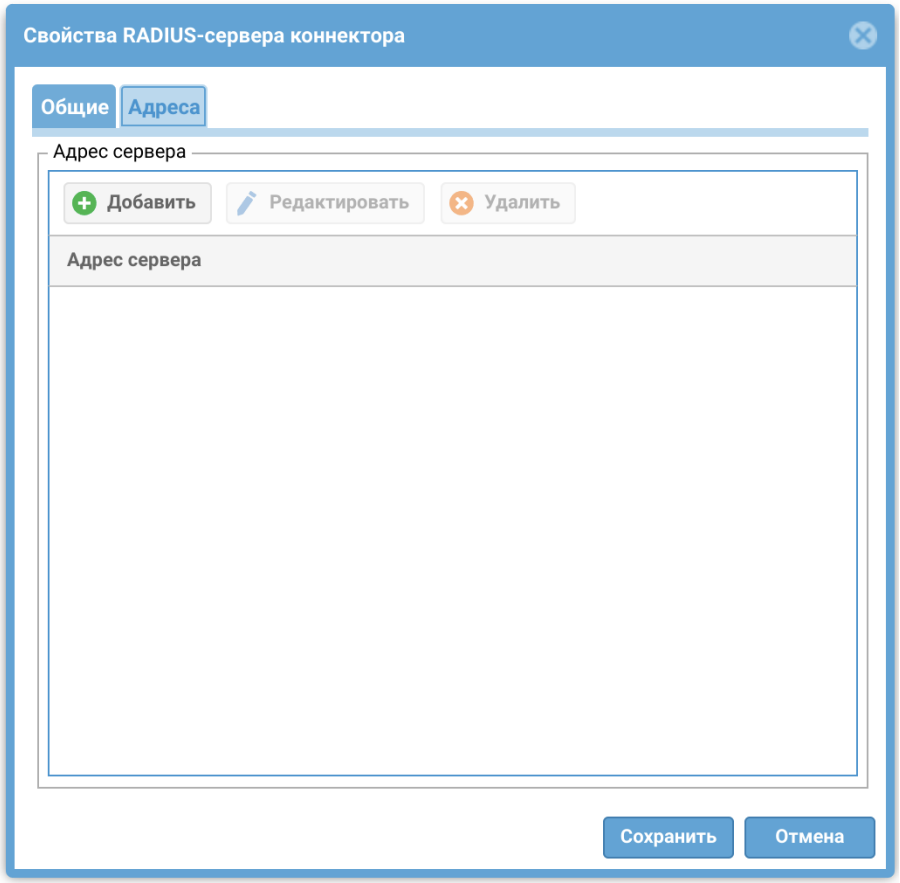
In the DCFW's admin web console, go to the **Settings → Users and devices → UserID agent connectors** section, click **Add** on the dashboard, and select **RADIUS server** as the type of connector. Then specify the following information:



- **Enabled**: enable/disable receiving logs from the source.
- **Name**: source name.
- **Description**: an optional description of the source.
- **Expiration time (sec.)**: the period of time that, when expires, leads to user's session termination, i.e. deleting their information from the DCFW's cache. The default value is 2700 seconds (45 minutes).
- **Authentication profile**: the authentication profile used to search for a user found in RADIUS accounting logs.
- **Attribute for name**: the radius attribute type number where the username resides. The default value is 1.
- **Attribute for groups**: the radius attribute type number where the user's group resides; the group is not verified by default.

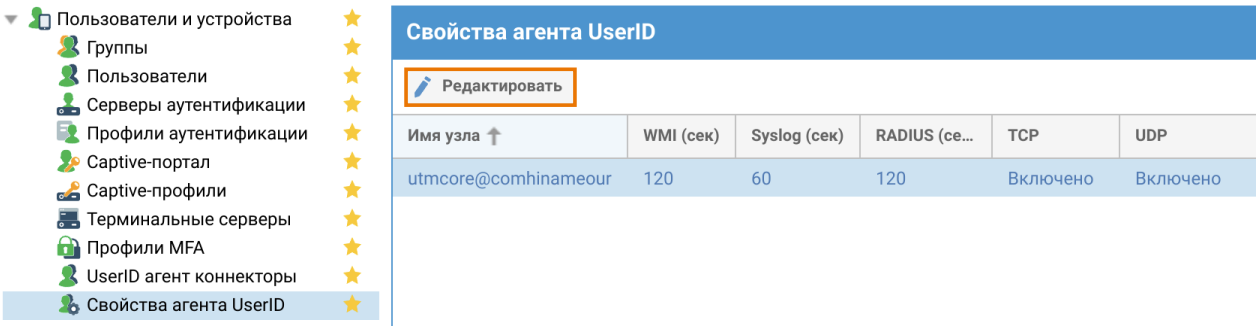
- **Default domain**: the name of a domain in which a user will be searched for in case the request does not indicate which domain they belong to.
- **Master node secret**: the pre-shared key used by the RADIUS protocol for authentication.

The **Addresses** tab is used to specify the host addresses (NAS servers) from which the UserID agent will receive RADIUS accounting events:

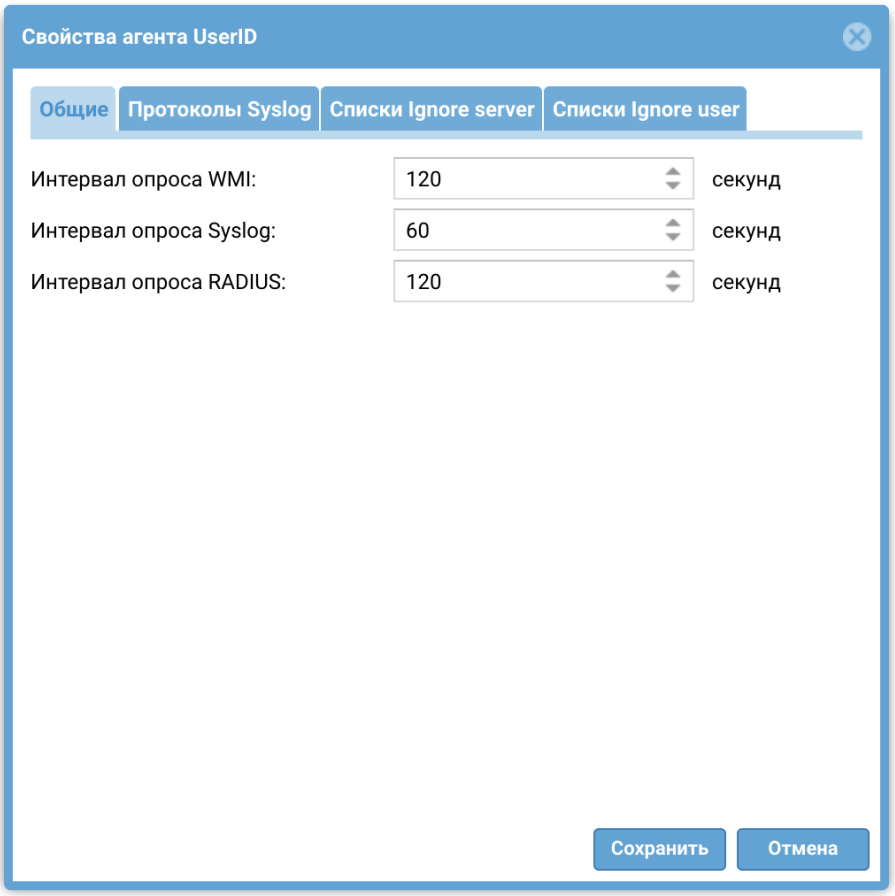


Configuring UserID Agent

The UserID agent's general settings can be configured in the **Settings → Users and devices → UserID agent properties** section. You need to click the **Edit** button on the dashboard:

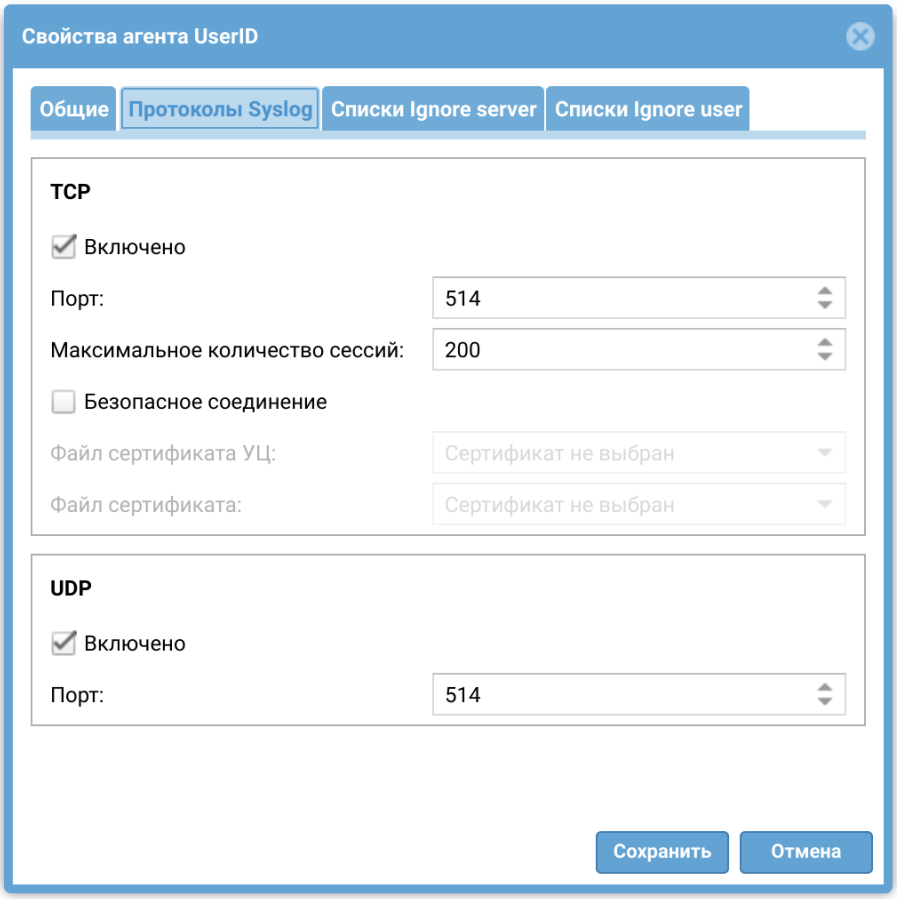


The **General** tab configures the data polling intervals:



- **WMI interval**: the period for Active Directory servers polling. The default value is 120 seconds.
- **Syslog interval**: the period for polling the database to search for session start / end events for syslog source users. The default value is 60 seconds.
- **Radius interval**: the period for polling the database to search for user session start / end events in the RADIUS log. The default value is 120 seconds. (This option is available starting from software version 7.2.0 and up).

The **Syslog protocols** tab allows to configure the syslog server connection parameters.



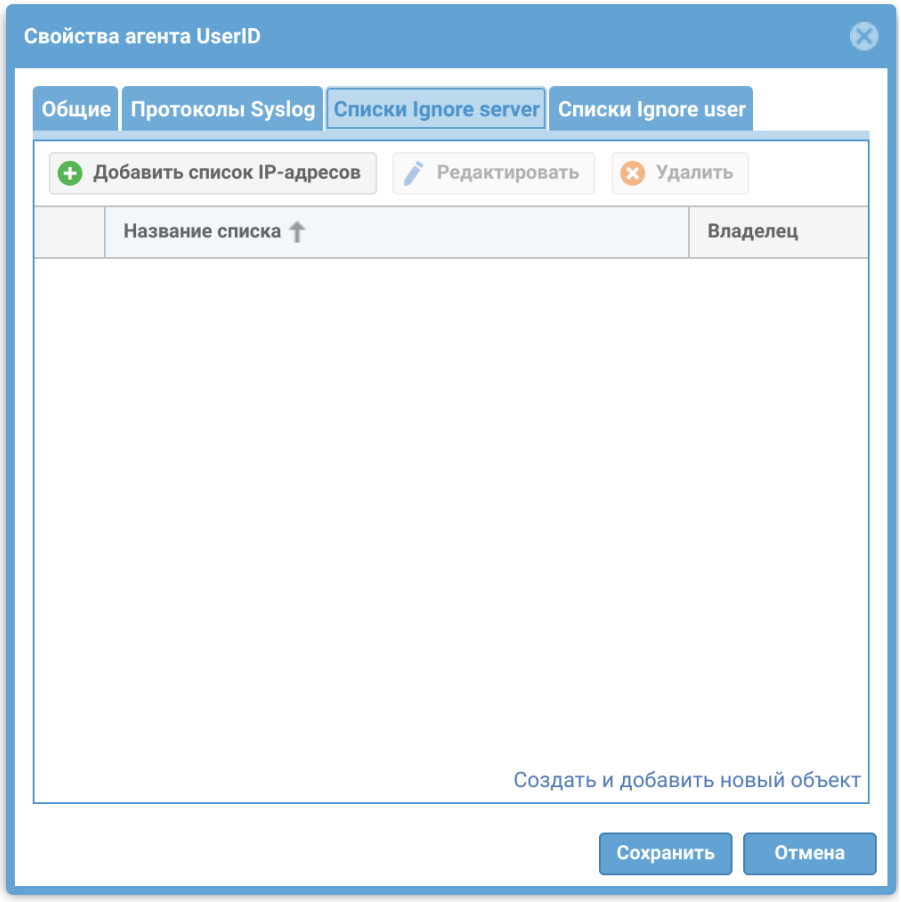
For the TCP protocol:

- **Enabled**: enabling/disabling the TCP protocol for receiving syslog logs.
- **Port**: the port number used to collect syslog events. The default port is 514.
- **Max session number**: the maximum number of concurrently connected devices that can be used to send messages.
- **Secure connection**: enabling disabling data flow encryption.
- **CA certificate file**: the certificate of the certification authority that is used to establish secure connections.
- **Certificate file**: the certificate created by a user and signed by a certification authority.

For the UDP protocol:

- **Enabled**: enabling/disabling the UDP protocol for receiving syslog logs.
- **Port**: the port number used to collect syslog events. The default port is 514.

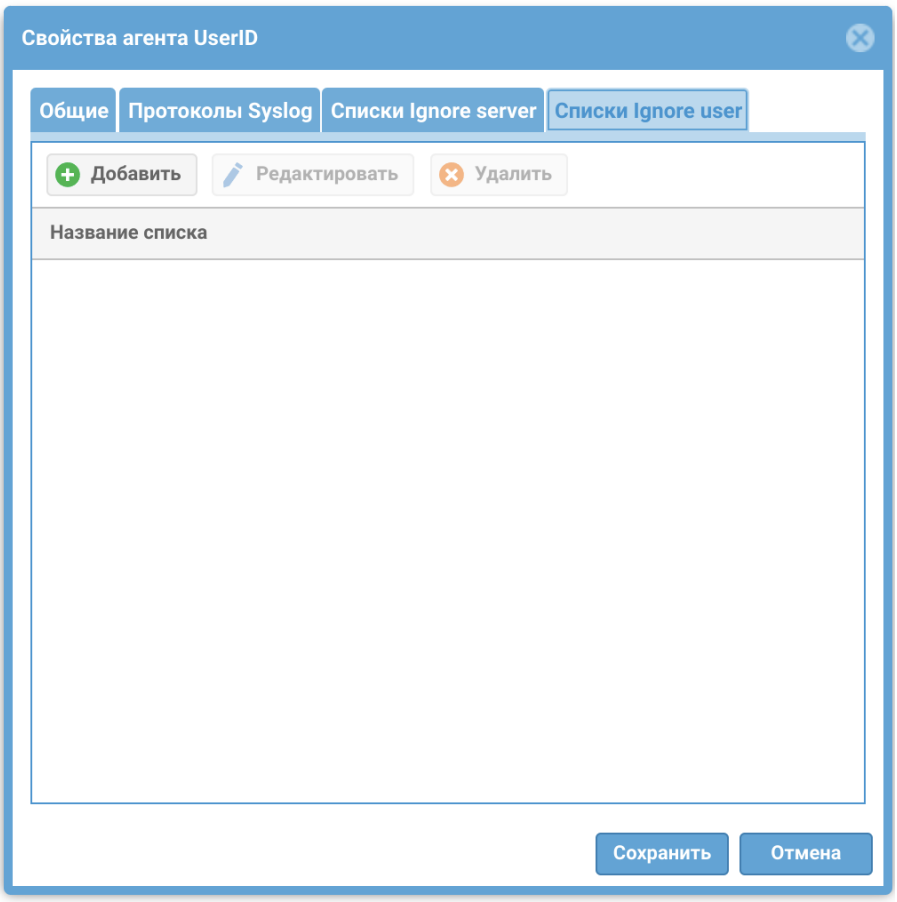
The **Ignore server list** tab allows you to specify the lists of IP addresses the events of which will be ignored by the UserID agent. Entries about ignored sources will appear in the UserID log:



You can create the list under the **Libraries → IP addresses** section, or do it when configuring the UserID agent (by clicking the **Create and add new object** button). For more details about how to create and configure IP address lists, see IP Addresses.

This setting is global and applies to all sources.

The **Ignore user list** tab allows you to specify the names of users whose events will be ignored by the UserID agent. The search is based on the Common Name (CN) of the AD user:



This setting is global and applies to all sources. A record about the ignored user appears in the UserID log.

Important! When specifying a name, you can use the asterisk (*), but only at the end of a string.

> **Note**
> **When DCFW connects to Log Analyzer, the UserID agents configured on both devices can operate simultaneously. The device agents will run independently of each other. UserID agent log events received by DCFW, as well as other log events, will be sent to LogAn.**

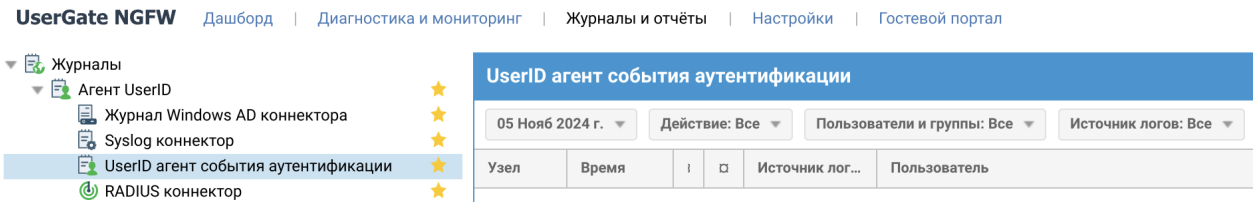
Logging

The UserID agent periodically communicates with configured data sources. Received events are stored in a service database without any changes. The contents of this database can be viewed in the following logs:

- Windows AD connector log;

- Syslog connector;
- RADIUS connector.

You can view them in the DCFW admin web console by navigating to the **Logs and reports → Logs → UserID agent** section.



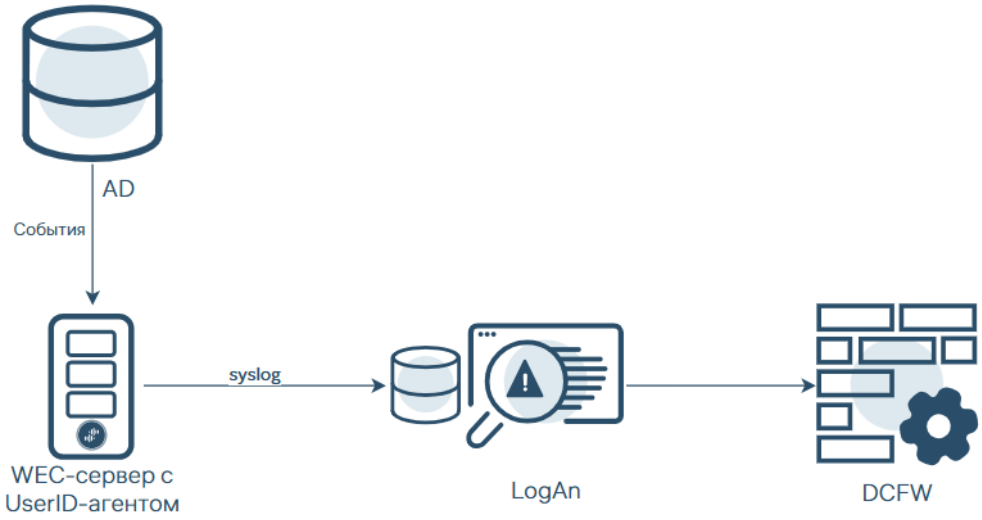
The UserID agent periodically communicates with the service database to extract the username, SID, domain, IP address, and group lists information from event records. The results of event records processing are registered in the "UserID agent auth events" log. This log is accessible in the same section: **Logs and reports → Logs → UserID agent**.

The descriptions of data source logs and UserID agent are provided in the UserID agent article of the "Logs and reports" section.

A description of the UserID log export formats is provided in the Appendix, in the Description of Log Formats section.

UserID agent for AD/WEC

The UserID agent for the AD/WEC is installed on the domain controller server or the WEC (Windows Event Collector) server. The agent reads the user identification information from the Windows security logs and transmits it to the UserID collector on the UserGate LogAn device in the `syslog` format.



Main Properties

The main UserID agent features for the AD/WEC:

- operation as a service;
- configuring the working parameters in the configuration file;
- reading the Windows security logs and sending the user data to the UserID collector via `syslog`;
- log file maintenance and rotation. The ability to enable or disable logging debug mode.

Installation

The UserID agent for AD/WEC is supplied as an installation file.

To install the UserID agent:

1. Download the latest version of the UserID agent from the official UserGate website.

2. Unzip the archive and run the *.msi installation file.

3. After installation is complete, navigate to the agent's working directory (by default: C:\Program Files (x86)\UserGate\useridagent) and edit the `useridagent.cfg` configuration file to include your network settings. For more information on the configuration file settings and format, see the Configuration section.

4. Restart the `UserIDAgent` service using the built-in Services application (Windows Services).

Configuration

You can configure the following configuration file settings:

- `ServerAddress`: the IP address of the UserGate DCFW interface at which the UserID agent transmits events.
- `EventFileNames`: names of logs to read. Default value: Security.
- `MaxLogSize`: maximum log file size in MB. Default value: 10.
- `EventIDs`: numbers of the events to be forwarded. For example, 4624, 4634 (network logon and network logoff event IDs).
- `DebugLogs`o: enables or disables debug mode. 0: basic events are logged, 1: an extended list of events is logged. Default value: 1.
- `UserExclude`: list of users for whom events should not be collected.
- `NetworkList`: list of subnets for which UserID events are collected.
- `GatewayList`: list of gateways (LogAn servers) to which found events are sent.

Configuration file syntax:

- The delimiter between parameters in the lists is a comma; spaces after the comma are ignored.
- Lines beginning with ";" and "#" are comments.
- The `[default]` section is optional, but if used the section must retain the specified name. The list of subnets and gateways should be specified in other sections, the names of which can be arbitrary. A new section begins with a line containing the character "`[`".
- The configuration file must have at least one section with one route and one gateway.

Configuration example:

```
[default]
ServerAddress=192.168.0.1:514
MaxLogSize=10
EventIDs=4634, 4624
EventFileNames=Security
```

```
DebugLogs=1
UserExclude=adm_,sys_
[net1]
NetworkList=192.168.30.0/24,172.30.250.0/24
GatewayList=192.168.45.1:514,192.168.45.2:514
[net2]
NetworkList=192.168.200.0/24,10.10.0.0/16
GatewayList=192.168.45.4:514,192.168.45.5:514
```

Example of minimal configuration:

```
[net1]
NetworkList=192.168.30.0/24
GatewayList=192.168.45.1
```

Logging

The UserID agent records information about events in the `uidagent.log` file, the size of which is controlled by the `MaxLogSize` parameter. When the limit is reached:

- the current log file is saved with the `*.bak` extension, replacing the previous version;
- writing of a new log file begins.

To store all service operation records, we recommend setting up external copying of log files using specialized services.

NETWORK POLICIES

General Information

The **Network policies** section contains the following subsections:

- Firewall
- NAT and routing

- Load balancing
- Traffic shaping.

Using network policies, the administrator can configure the required Internet access for the users, publish internal resources to the Internet, and manage the bandwidth for specific services and applications.

> **Note**
> **The rules created in these sections are applied top to bottom as they are listed in the console. Only the first rule matching the conditions is triggered. This means that more specific rules must be placed higher in the list than more general ones.**

To enable Internet access for the users, follow these steps:

Name	Description
Step 1. (Optional) Create a NAT rule.	If it is necessary to replace traffic addresses. See the NAT and Routing section.
Step 2. Create a firewall rule allowing access.	See the Firewall section.

To publish an internal resource to the Internet:

Name	Description
Step 1. Create DNAT rule.	See the DNAT Rules section.

To provide Internet access via an alternate provider for a specific service or address:

Name	Description
Step 1. Create a policy-based routing rule.	See the Policy-Based Routing section.

To block or allow a specific type of traffic that passes through UserGate:

Name	Description
Step 1. Create a firewall rule.	See the Firewall section.

To distribute traffic between multiple internal servers:

Name	Description
Step 1. Create a load balancing rule.	See the Load Balancing section.

To limit the bandwidth for a specific service or application:

Name	Description
Step 1. Create a traffic shaping rule.	See the Traffic Shaping section.

Firewall

Using firewall rules, the administrator can allow or deny any type of transit network traffic that passes through UserGate DCFW. Source/destination zones or IP addresses, users, groups, services can all be used as conditions for the rules.

Firewall rule trigger events are displayed in the traffic log (**Logs and reports → Traffic**) when **Logging** is enabled in the rule settings.

Note

The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.

Note

The *Negate* checkbox changes the condition to the opposite, which corresponds to a Boolean NOT (negation).

Note

If there are no rules created, all transit traffic via DCFW is blocked.

IDPS profiles and/or application profiles (L7) that contain certain signature sets can be added to firewall's allow rules. Once traffic hits the first allow rule of the firewall, the data flow begins to be analyzed by the signatures of the IDPS and/or L7 profiles. When these signatures are triggered, the action configured in the rule is applied to the traffic, and the corresponding entry is made in the logs (**Traffic** for applications and **IDPS** for the IDPS), if the **Logging** option was enabled in the profiles.

To create a firewall rule, go to **General settings → Network Policies → Firewall**, click **Add** and specify the required rule parameters.

For a rule to be triggered, all conditions specified in the rule's settings must match.

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.
Description	A description of the rule.
Action	**Deny**: blocks the traffic. **Allow**: allows the traffic
Applications Profile	An application profile created in advance in **General settings → Libraries → Application Profiles**. An applications profile contains a set of application signatures intended for use in firewall rules for traffic analysis at Layer 7 of the OSI model. For more details on creating and configuring applications profiles, see the Applications Profiles section. **Important!** **An applications profile is an additional setting that activates traffic analysis at Layer 7 of the OSI model. It can only be used in firewall rules that allow traffic.**
IDPS profile	An IDPS profile created earlier in the **General settings → Libraries → IDPS profiles** section. An IDPS profile is a set of relevant signatures used for detecting intrusions and protecting certain services. For more details on creating and configuring IDPS profiles, see the IDPS Profiles section.

Parameter	Description
	Important! **An applications profile is an additional setting that activates an IDPS rule. It can only be used in firewall rules that allow traffic.**
Reject with	This parameter is available in rules that block traffic (with the **Deny** action selected). It can take one of the following values: • **Not selected**. • **Send ICMP host unreachable**: block the traffic and send an ICMP message. • **Send TCP reset**: block the traffic and send a TCP connection reset message. **Important!** To select the **Send TCP reset** action, a service that uses the TCP protocol (the **Service tab**) must be selected. • **Send TCP reset to both parties**: block the traffic and send a TCP connection reset message to the client and server
Scenario	The scenario that must be active for the rule to be triggered. For more details on how scenarios work, see the Scenarios section. **Important!** **A scenario is an additional condition. If the scenario was not triggered (one or more scenario triggers did not occur), the rule will not be triggered.**

Parameter	Description
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **Log all network packets**: every transmitted network packet will be logged. For this mode, it is recommended to enable the logging limit to prevent high device load. • **No**. Nothing will be logged
Source	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** **Lines with the '*' symbol in such lists do not work (they are ignored).** Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified
Users	The list of users or groups to which this rule is applied. The **Any**, <0>Unknown, and <0>Known user types can be used. To apply rules to specific users or **Known** users, user identification needs to be configured. For more details on user identification, see the Users and Devices section.
Destination	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic destination. The URL list must include only domain names.

Parameter	Description
	Important! **Lines with the '*' symbol in such lists do not work (they are ignored).** Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Service	The service type, such as HTTP or HTTPS
Time	The time periods when the rule is active
Usage	The trigger statistics for the rule: the total trigger count, the time of the first and last trigger, and triggers by application. To reset the trigger count, select the rules in the list and click **Reset hit counts**
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

NAT and Routing

In the **NAT and routing** section, the administrator can create NAT, DNAT, port forwarding, policy-based routing, and network mapping rules. UserGate DCFW supports NAT/DNAT for complex protocols that can use dynamic ports. FTP, PPTP, SIP, and H323 protocols are supported.

Trigger events for NAT, DNAT, port forwarding, policy-based routing, and network mapping rules are displayed in the traffic log (**Logs and reports → Traffic**) when **Logging** is enabled in the rule settings.

> **Note**
> **GeoIP cannot be used as the traffic source address of in NAT rules and as the destination address of traffic in NAT, DNAT, and port forwarding rules.**

NAT Rules

Generally, enabling Internet access for the users requires creating at least one NAT rule from the **Trusted** to the **Untrusted** zone.

> **Note**
> **The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.**

> **Note**
> **The *Negate* checkbox changes the condition to the opposite, which corresponds to a Boolean NOT (negation).**

To create a NAT rule, go to the **Network policies → NAT and routing** section, click **Add**, and provide the desired settings.

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.
Description	A description of the rule.
Type	Select **NAT**.
SNAT IP address (external IP)	Explicitly sets the IP address with which the source address will be replaced. This makes sense if there are multiple IP addresses assigned to the destination zone's interfaces. If left empty, the system will use an arbitrary address from the list of available IP addresses assigned to the destination zone's interfaces. A range of IP addresses may be specified, for example:

Parameter	Description
	192.168.10.10-192.168.10.20 In this case, DCFW will use all addresses from the range for Source NAT. It is recommended to specify a SNAT IP explicitly to improve firewall performance.
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **No**. Nothing will be logged.
Source	The zone, IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The **Negate** checkbox does not affect rule processing, when MAC addresses are used. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.
Destination	The zone, IP address lists, or destination URL lists of the traffic. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.
Service	The service type, such as HTTP, HTTPS or other.

Parameter	Description
Usage	The trigger statistics for the rule: the total trigger count and the time of the first and last trigger. To reset the trigger count, select the rules in the list and click **Reset hit counts**.
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

> **Note**
> **It is recommended to create general NAT rules, such as NAT from the local network (normally the Trusted zone) to the Internet (normally the Untrusted zone), and access restrictions by user, service, and application using firewall rules.**

DNAT Rules

DNAT rules are normally used to publish internal network resources to the Internet.

> **Note**
> **The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.**

> **Note**
> **The *Negate* checkbox changes the condition to the opposite, which corresponds to a Boolean NOT (negation).**

To create a DNAT rule, go to the **Network policies → NAT and routing** section, click **Add**, and provide the desired settings.

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.

Parameter	Description
Description	A description of the rule.
Type	Select **DNAT**.
SNAT IP address (external IP)	Explicitly sets the IP address with which the source address will be replaced. If SNAT IP is not specified, the source address will be replaced with the address of the DCFW interface from which the packet was sent. A range of IP addresses may be specified, for example: 192.168.10.10-192.168.10.20 **Important!** To have the source address replaced with the specified address, the **Enable SNAT** checkbox must be set on the **DNAT** tab.
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **No**. Nothing will be logged.
Source	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The **Negate** checkbox does not affect rule processing, when MAC addresses are used. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Destination	One of the external IP addresses of DCFW, which is accessible from the Internet and represents the destination for external client traffic.

Parameter	Description
	Important! The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.
Service	The type of service to publish, such as HTTP. If not specified, all services will be published. **Important!** Services that use the following ports may not be published as these ports are reserved for UserGate's internal services: 2200, 8001, 4369, 9000-9100.
DNAT target IP (published server IP)	The IP address of a computer in the local network that is being published to the Internet.
Enable SNAT (change source IP to UserGate IP)	If enabled, DCFW will replace the source address in the packets from the external network with its own IP address.
Usage	The trigger statistics for the rule: the total trigger count and the time of the first and last trigger. To reset the trigger count, select the rules in the list and click **Reset hit counts**.
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

Port Forwarding Rules

Port forwarding rules work similarly to DNAT rules, except that they allow you to change the port number on which an internal service is published. To create a port forwarding rule, go to the **Network policies → NAT and routing** section, click **Add**, and provide the desired settings.

> **Note**
>
> **The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.**

> **Note**
> **The *Negate* checkbox changes the condition to the opposite, which corresponds to a Boolean NOT (negation).**

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.
Description	A description of the rule.
Type	Select **Port forwarding**.
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **No**. Nothing will be logged.
Source	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The **Negate** checkbox does not affect rule processing, when MAC addresses are used. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Destination	The zone, IP address lists, or destination URL lists of the traffic. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored).

Parameter	Description
	Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.
Port forwarding	Port overriding for the published services: • **Original destination port**: the TCP/UDP port number to which the users send requests. **Important!** The ports 2200, 8001, 4369, 9000-9100 may not be used, as they are used by the internal DCFW services. • **New destination port**: the TCP/UDP port number of the internal server being published to which user requests will be forwarded.
DNAT target IP (published server IP)	The IP address of a computer in the local network that is being published to the Internet.
Enable SNAT (change source IP to UserGate IP)	If enabled, DCFW will replace the source address in the packets from the external network with its own IP address.
Usage	The trigger statistics for the rule: the total trigger count and the time of the first and last trigger. To reset the trigger count, select the rules in the list and click **Reset hit counts**.
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

Policy-Based Routing

Policy-based routing rules are normally used to define a specific route to the Internet for certain hosts and/or services. For example, an organization that uses two Internet providers may need to route all HTTP traffic via provider 1 and all the rest via provider 2. To do that, it would set the Internet gateway of provider 2 as the default gateway and configure a policy-based routing rule for HTTPS traffic via the gateway of provider 1.

> **Note**
> **PBR rules do not replace NAT rules or affect how they work. For network address translation, place a corresponding NAT rule after a PBR rule.**

> **Note**
> **The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.**

> **Note**
> **The *Negate* checkbox changes the condition to the opposite, which corresponds to a Boolean NOT (negation).**

To create a policy-based routing rule, go to the **Network policies → NAT and routing** section, click **Add<0>, and provide the desired settings.**

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.
Description	A description of the rule.
Type	Select **Policy-Based Routing**.
Gateway	Select one of the existing gateways. You can add a gateway in the **Network → Gateways** section. **Important!** The selected gateway may belong to a specific virtual router.
Scenario	The scenario that must be active for the rule to be triggered. For more details on how scenarios work, see the Scenarios section. **Important!** A scenario is an additional condition. If the scenario was not triggered (one or more scenario triggers did not occur), the rule will not be triggered.

Parameter	Description
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **No**. Nothing will be logged.
Source	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The **Negate** checkbox does not affect rule processing, when MAC addresses are used. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Users	The list of users or groups to which this rule is applied. The **Any**, <0>Unknown, and <0>Known user types can be used. To apply rules to specific users or **Known** users, user identification needs to be configured. For more details on user identification, see the Users and Devices chapter.
Destination	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic destination. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.

Parameter	Description
	• The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Service	The service type, such as HTTP, HTTPS or other.
Usage	The trigger statistics for the rule: the total trigger count and the time of the first and last trigger. To reset the trigger count, select the rules in the list and click **Reset hit counts**.
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

Network Mapping

Network mapping rules allow substitution of the source or destination network address. This is usually required when there are multiple networks with identical addressing, such as 192.168.1.0/24, that need to be merged into a single routed network. Without network address substitution, this kind of merge would be impossible. Network mapping changes only the network address, leaving the host address as is: for example, if source network 192.168.1.0/24 is substituted with 192.168.2.0/24, host 192.168.1.1 will change to 192.168.2.1.

> **Note**
> **The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.**

To create a **Network mapping** rule, go to the **Network policies → NAT and routing** section, click **Add**, and provide the desired settings.

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.

Parameter	Description
Description	A description of the rule.
Type	Select **Network mapping**.
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **No**. Nothing will be logged.
Source	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The **Negate** checkbox does not affect rule processing, when MAC addresses are used. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.

Parameter	Description
Destination	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic destination. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Service	The service type, such as HTTP, HTTPS or other.
Network Mapping	Configure the network substitution settings. **Direction**: • Input, replace destination network address: destination IP addresses in the traffic that matches the rule conditions will be substituted. The network address is substituted with the one specified in the **New IP network/mask** field. • Output, replace source network address: source IP addresses in the traffic that matches the rule conditions will be substituted. The network address is substituted with the one specified in the **New IP network/mask** field • **New IP network/mask**: the network address that gets substituted for the original one.
Usage	The trigger statistics for the rule: the total trigger count and the time of the first and last trigger. To reset the trigger count, select the rules in the list and click **Reset hit counts**.
History	The time the rule was created and last changed as well as the related event log entries, such as rule added, rule updated, rule list position changed etc.

Load Balancing

DCFW supports load balancing for various services within the local network. Load balancing can be provided for:

UserGate DCFW supports traffic balancing between different services. Balancing can be used in the following scenarios:

- for internal servers published to the Internet using DNAT;
- for internal servers not published to the external network;

The load balancer accepts requests directed to the virtual server's IP address and distributes them among the IP addresses of real servers using various balancing algorithms.

To create a TCP/UDP load balancing rule, go to **General settings → Network policies → Load balancing**, click **Add**, select <0>TCP/UDP load balancer, and specify the required parameters.

Parameter	Description
Enabled	Enable or disable the load balancing rule.
Name	The name of the balancing rule
Description	A description of the balancing rule
Virtual server IP address	Select an address from the list of IP addresses assigned to the node's network interfaces. You can add additional IP addresses to the desired interface if necessary.
Port	The port for which load balancing is to be performed
Protocol	Protocol for which load balancing is required (TCP or UDP).
Scheduler	Select a method for distributing traffic to real servers: • **Round robin**: each new connection is forwarded to the next server in the list, evenly loading all servers. • **Weighted round robin**: works similarly to the round robin method, but the load on real servers is weighted, allowing for load distribution based on the performance of each server.

Parameter	Description
	• **Least connections**: a new connection is passed to the server which currently has the least number of connections. • **Weighted least connections**: works similar to least connections but the load is distributed between the real servers according to their assigned weight factors, which allows each server's performance to be taken into account.
Real servers	Add a pool of real servers between which the traffic will be distributed. For each of the servers, provide these settings: • **IP address** of the server. • **Server port**: the port to which user requests should be forwarded. • **Weight**: a parameter for unevenly distributing traffic to real servers for the weighted round robin and weighted least connections balancing modes. The greater the weight, the higher the server load. • **Mode**: ◦ **Gateway**: routing is used to redirect traffic to the virtual server. ◦ **Masq**: DNAT is used to forward the traffic to the virtual server. ◦ **Masq with SNAT**: similar to Masq, but with UserGate DCFW substituting the source IP address with its own. **Important!** **Since the load balancer does not change packets headers in the Gateway mode, the reverse traffic from the real server needs to be set up via routing. It means that the gateway address for the reverse traffic must be different from the UserGate DCFW address.**
Fallback	The fallback mode is used when none of the real servers is available. To activate fallback, enable it and provide these settings: • **IP address** of the server. • **Port** of the server. the server port to which user requests will be forwarded.

Parameter	Description
	• **Mode**. There are three options: ◦ **Gateway**: routing is used to forward the traffic to the virtual server ◦ **Masq**: DNAT is used to forward the traffic to the virtual server. ◦ **Masq with SNAT**: similar to Masq, but with UserGate DCFW substituting the source IP address with its own.
Monitoring	The monitoring mechanism checks the availability of real servers. Unavailable servers are automatically removed from the load balancing pool.
Aggregation mode	Real server monitoring mode: • **ping**: checks node availability using the ping utility. • **connect**: check if the node is up and running by establishing a TCP connection to a specific port. • **negotiate**: check node health by sending a certain HTTP or DNS request and comparing the response against the expected one. To configure this mode, select the service type (**HTTP** or **DNS**) and specify the **Request** and **Expected response** strings. Here is an example for an HTTP request: ◦ **Request**: `/robots.txt`; ◦ **Expected response**: `Disallow: /bin/`. The request string here points to the real server path that will be used in the HTTP request. The expected response string contains a fragment of the response webpage
Check interval	The time interval for the periodic health check
Check timeout	Check response timeout.
Max failures	The number of failed health check attempts after which a real server will be considered unhealthy and excluded from load balancing list.

Note

The balancing rules have a higher priority than NAT, DNAT, routing rules and are applied before them.

Traffic Shaping

Traffic shaping rules are used to limit the bandwidth for certain users, hosts, services, or applications.

> **Note**
> **The rules are applied top to bottom in their listing order. Only the first rule in which all conditions are matched is applied. This means that more specific rules must be placed higher in the list than more general ones. To change the order in which the rules will be applied, use the Up/Down and Top/Bottom buttons or drag and drop the rules with the mouse.**

> **Note**
> **The "Negate" checkbox changes the condition to the opposite, which corresponds to a Boolean NOT (negation).**

To create a traffic shaping rule, go to the **Network policies → Traffic shaping** section, click **Add**, and provide the desired settings.

Parameter	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.
Description	A description of the rule.
Bandwidth pools	Select one of the bandwidth pools. A bandwidth pool can optionally change the priority tags of DSCP traffic. For instructions on how to create more bandwidth pools, see the Bandwidth Pools section.
Scenario	The scenario that must be active for the rule to be triggered. For more details on how scenarios work, see the Scenarios section.

Parameter	Description
	Important! A scenario is an additional condition. If the scenario was not triggered (one or more scenario triggers did not occur), the rule will not be triggered.
Logging	Logs traffic information when the rule is triggered. The available options are: • **Log session start**: only the session start (first packet) will be recorded in the traffic log. This is the recommended logging option. • **Log all network packets**: every transmitted network packet will be logged. For this mode, it is recommended to enable the logging limit to prevent high device load. • **No**. Nothing will be logged.
Source	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic source. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified. • The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Users	The users or user groups to which this rule will be applied.
Destination	The zone, IP address lists, Geo-IP address lists, or URL lists of the traffic destination. The URL list must include only domain names. **Important!** Lines with the '*' symbol in such lists do not work (they are ignored). Every 5 minutes DCFW resolves domain names into IP addresses and stores the result in the internal cache for the DNS record's time-to-live (TTL). When the TTL expires, DCFW automatically updates the IP address value. **Important!** The traffic processing logic is as follows: • The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.

Parameter	Description
	• The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.
Service	The service type, such as HTTP, HTTPS or other.
Applications	The list of applications for which bandwidth needs to be limited.
Time	The time when this rule will be active.

VPN SETTINGS

VPN General terms

VPN (Virtual Private Network) is a generic name for technologies that make it possible to create logical networks (tunnels) on top of public networks to provide communications security.

To create a VPN, at least two network devices are needed that can identify each other and encrypt the data flow between them.

Types of VPN Connection

UserGate DCFW allows you to create the following types of VPN connections:

- **Site-to-Site VPN** connections. In this case, one host works as a VPN server and another as a VPN client. This kind of server-to-server connection allows you to consolidate corporate offices into a single logical network.
- **Remote Access VPN**. In this case, DCFW UserGate works as a VPN server and user devices as VPN clients.

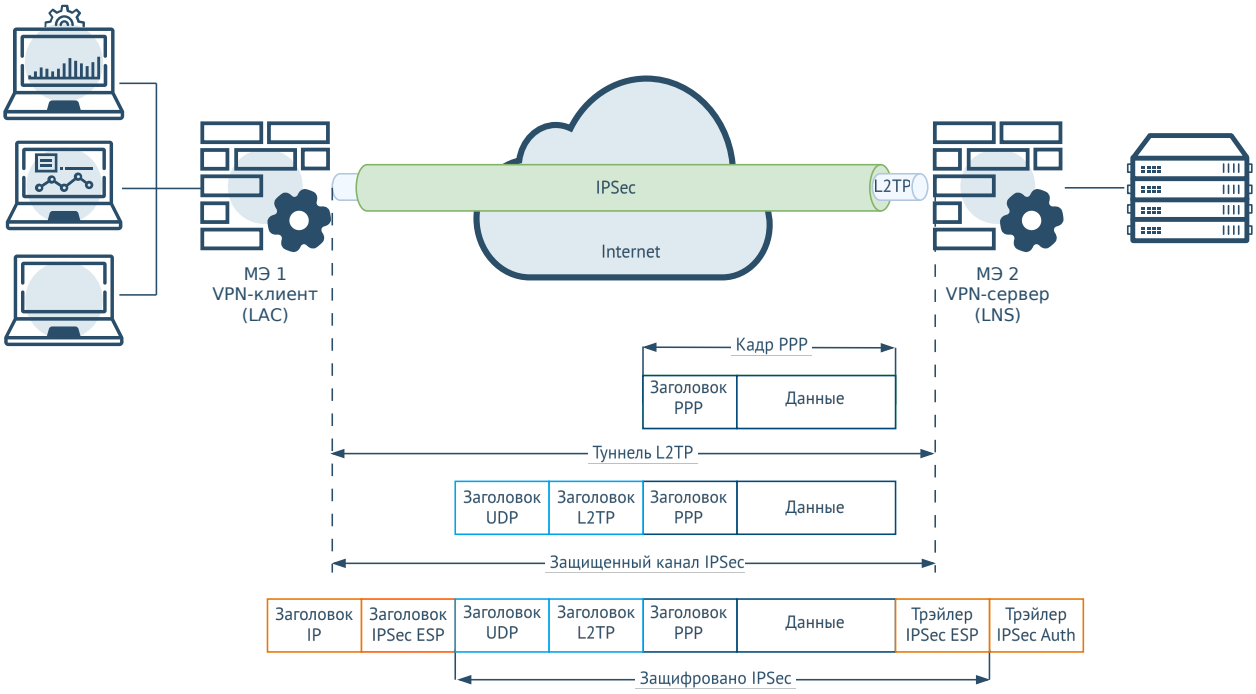
Secure VPN Tunneling Options

Secure VPN tunnels can be created using L2TP/IPsec(IKEv1), IPsec(IKEv2), IPSec(IKEv1), GRE/IPsec protocols.

L2TP/IPsec VPN

With an **L2TP/IPsec** VPN, a tunnel is created using L2TP (RFC 3931) protocol where network-layer packets are transmitted inside PPP frames. Since L2TP itself does not provide strict authentication, confidentiality, and integrity of the data being transmitted, the IPsec (RFC 6071) group of protocols is used for those purposes.

The L2TP tunnel is created inside a secure IPsec link, and to establish it, you first need to create a secure IPsec connection between the hosts. In this case, IPsec works as a transport and uses ESP (Encapsulating Security Payload) to encrypt L2TP packets.

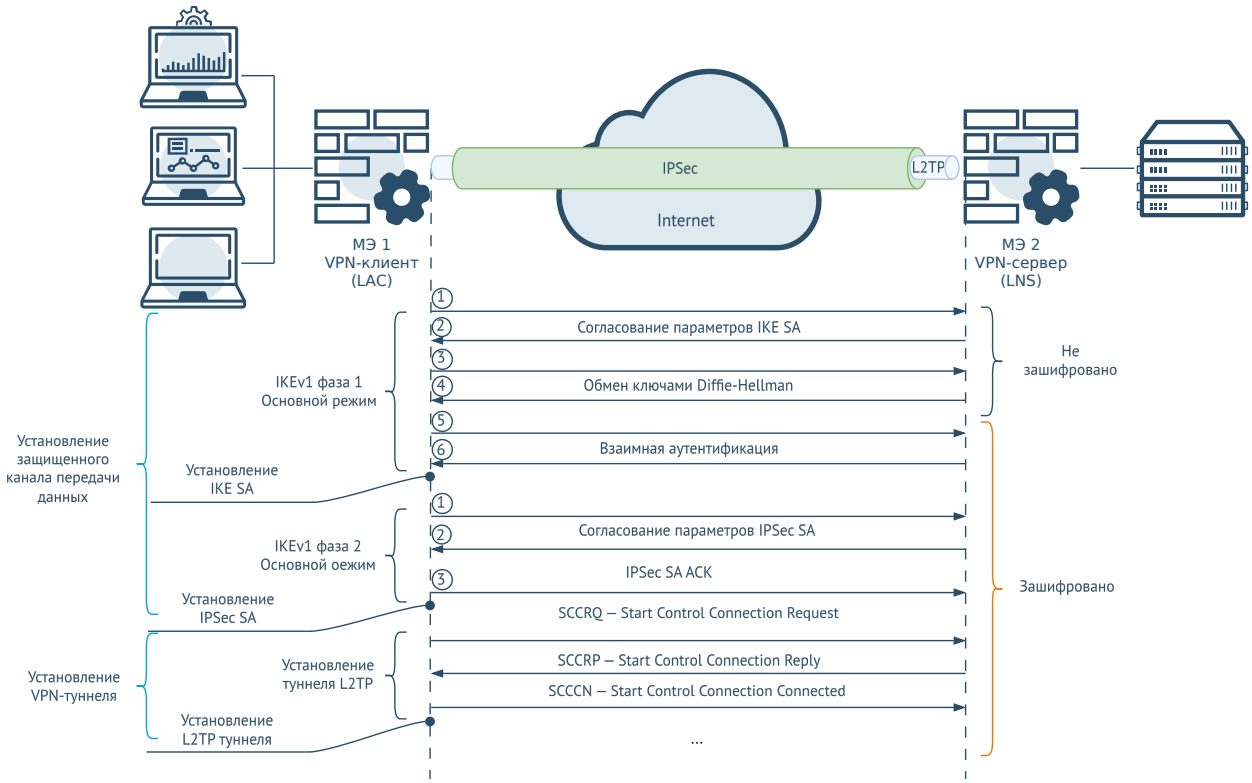


VPN has two encapsulation levels: internal L2TP encapsulation and external IPsec encapsulation. The internal level has L2TP and UDP headers in addition to the PPP frame. The external level adds an IPsec ESP header and trailer. An IPsec Auth trailer provides message integrity check and authentication.

The process of creating a VPN comprises these main steps:

1. Establish a secure data link

2. Establish a VPN tunnel.



Secure data link establishment

To establish a secure data link, the IPsec group of protocols is used.

IPsec has three underlying protocols:

- Authentication Header (AH)
- Encapsulating security payload (ESP)
- Internet Key Exchange (IKE).

Authentication Header (AH) ensures the integrity of the transmitted data, authentication of the data source, and protection from retransmission. AH does not provide confidentiality for the data in transit because it does not perform encryption. The IP protocol number for AH is 51.

Encapsulating security payload (ESP) uses encryption to ensure the confidentiality of the transmitted data and also supports data integrity and data source authentication. The IP protocol number for ESP is 50.

Internet Key Exchange (IKE) is a service data exchange protocol for negotiating an establishing a security association (SA). A security association includes a set of secure connection parameters that can be used by both sides of the connection for mutual authentication and the encryption of transmitted data. IKE uses the UDP port 500.

The IPsec has two modes of operation:

- Tunnel mode
- Transport mode.

In the tunnel mode, IPsec encrypts the entire original IP packet along with its header. Then it is encapsulated inside an additional packet that has its own header. The tunnel mode is used when two private networks transmit data via an insecure public network.

In the transport mode, only the payload of the IP packet is encrypted, and the original IP header is kept with some extra information added to it. The transport mode is used when the two hosts already have an IP connection, but that connection does not provide security for the transmitted data.

When a VPN is created using L2TP/IPsec, the tunnel between the two hosts is created using the L2TP protocol, and IPsecis to provide security for the data link. In this case, IPsec operates in the transport mode, and the negotiation of the security association and establishment of the secure link is done using the IKE protocol (IKEv1) in two phases.

In **phase 1**, the neighboring hosts are mutually authenticated, an IKE SA is negotiated, and a secure service link is established between the hosts for IKE data exchange.

The negotiated parameters of an IKE SA are:

- Hash algorithms (MD5, SHA)

- Encryption algorithms (DES, 3DES, AES)
- Tunnel lifetime parameters
- Diffie-Hellman groups.

Pre-shared keys are used for authenticating the link peers.

IKEv1 phase 1 negotiation can be carried out in two modes:

- Main
- Aggressive.

In the main mode, the devices exchange six messages. During the first exchange (messages 1 and 2), the encryption and authentication algorithms are negotiated for IKE SA. The second exchange (messages 3 and 4) implements the Diffie-Hellman (DH) key exchange. After the second exchange, the IKE service on each device creates a master key to use for authentication. The third exchange (messages 5 and 6) authenticates the reporter and responder of the connection (identity checking) and the information is secured using the encryption algorithm established earlier.

In the aggressive mode, there are two message exchanges with three messages in total. In the first message, the reporter transmits information corresponding to messages 1 and 3 of the main mode — that is, the information on encryption and authentication algorithms as well as the DH key. The second message, transmitted by the responder, contains information corresponding to messages 2 and 4 of the main mode and also authenticates the responder. The third message authenticates the reporter and confirms the exchange.

The fewer number of messages allows for a quicker connection establishment in the aggressive mode, but the peer IDs are exchanged in plain text. The main mode is considered more secure because it encrypts the ID data.

The result of phase 1 is a successfully negotiated bidirectional IKE SA and the establishment of a secure service link. This channel will be used in phase 2 for negotiating the IPsec SA parameters for the main data channel.

In **phase2**, the secure service link established earlier in phase1 is used to negotiate an IPsec SA for the secure transmission of data over the IPsec link.

IKEv1 has one mode in phase 2 called quick mode.

The negotiated parameters of an IPsec SA are:

- Encryption algorithms (DES, 3DES, AES)
- Hash algorithms (MD5, SHA-1, SHA-2)
- SA lifetime parameters.

Based on the IPsec SA negotiation results, a secure data link is created operating in the IPsec transport mode with ESP encapsulation that has two unidirectional SAs to both sides of the link.

After this mode is set, the service link established in phase 1 does not disappear and is used to update the SA for the main channel.

The IPsec SA is terminated when the VPN is disconnected on one side of the connection or upon a timeout. A timeout occurs when the key lifetime has elapsed or the link lifesize has been reached. When the SA is terminated, the keys are deleted. If additional IPsec SAs are required for data transmission, a new IKE negotiation phase follows.

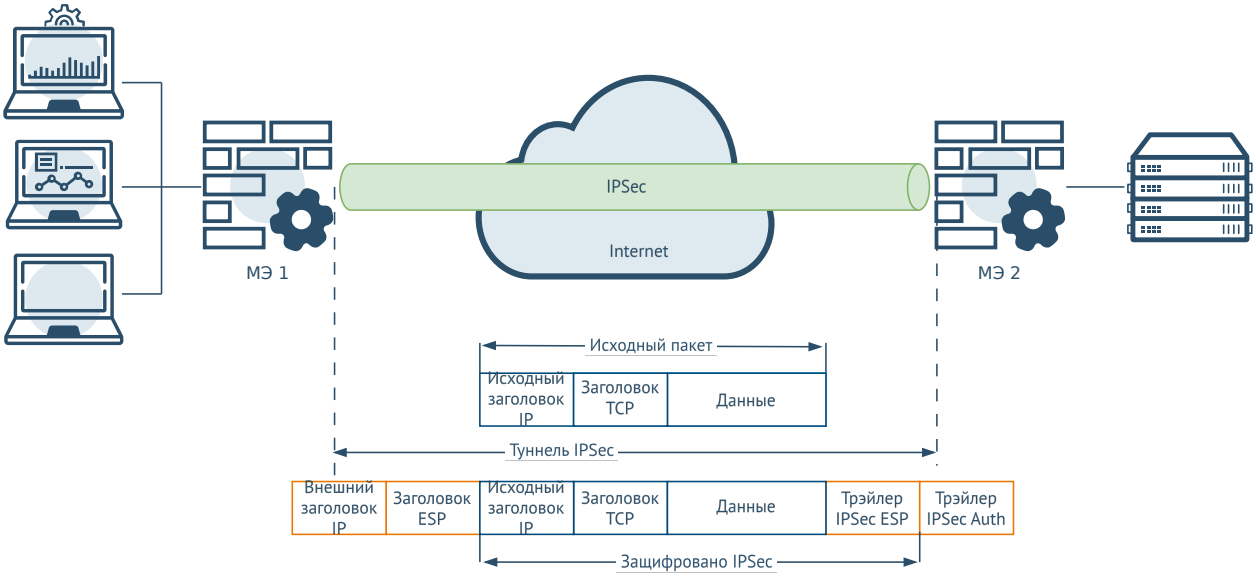
VPN tunnel establishment

At this stage, an L2TP tunnel is negotiated and established between the SA endpoints. The actual parameter negotiation occurs over a secure IPsec SA link. The L2TP protocol uses UDP port 1701.

When the VPN has been established, L2TP packets flowing between the endpoints are encapsulated using IPsec. Since the L2TP packet itself is wrapped into an IPsec packet, the original source and destination IP addresses are encrypted inside the outer packet. In addition, there is no need to open UDP port 1701 on the firewall between the endpoints because the inner packets are not processed until the IPsec data have been decrypted, which only happens at the tunnel endpoints.

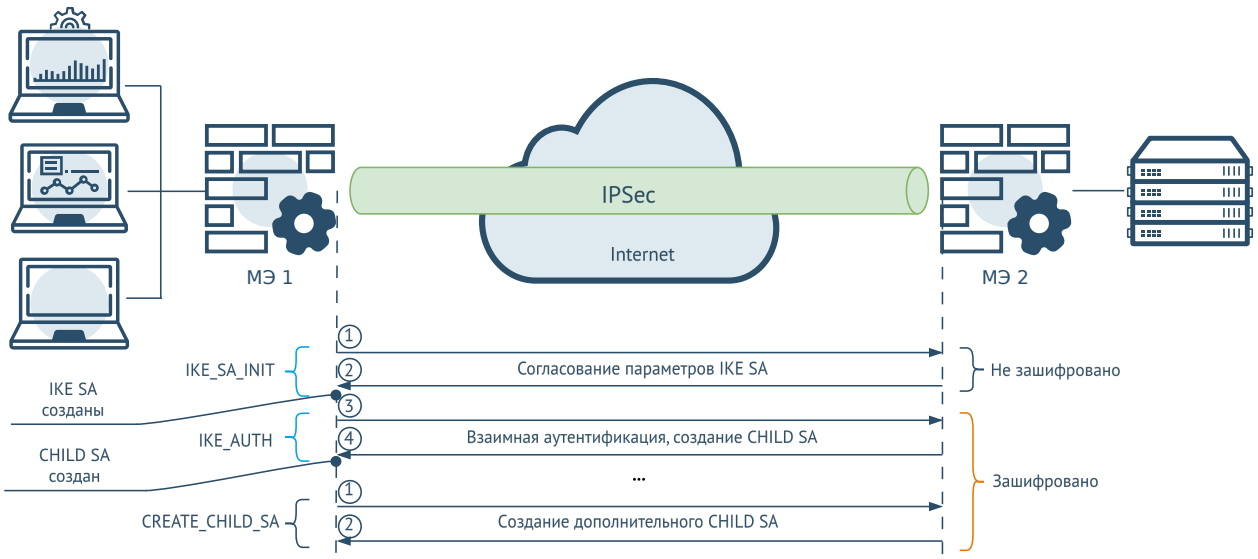
IPsec(IKEv2) VPN

When a VPN is created using **IPsec(IKEv2)**, a secure VPN tunnel is established only using the IPsec group of protocols together with IKEv2(RFC 7296, RFC 7427).



In this scenario, IPsec operates in the tunnel mode where the original IP packets are fully encapsulated and encrypted inside a new packet that has its own header and trailers.

Similar to IKEv1, IKEv2 also has a two-stage secure connection establishment process, but with fewer message exchanges.



The first stage is known as **IKE_SA_INIT**. In the two messages comprising an IKE_SA_INIT exchange between two neighboring hosts, IKE SA parameters are negotiated, and a secure service link is established. The negotiated parameters of an IKE SA are:

- Hash algorithms
- Encryption algorithms
- Diffie-Hellman key.

Every host generates a seed key (SKEYSEED) that is used afterwards to generate keys used in the IKE SA. All the subsequent IKE keys are generated using the SKEYSEED.

The second stage is called **IKE_AUTH**. At this stage, the neighboring hosts are authenticated. The two messages comprising an IKE_AUTH exchange are authenticated and encrypted in the context of the IKE SA created during the IKE_SA_INIT message exchange.

At the end of the second negotiation stage, a child security association (CHILD SA) is created under the IKE SA for secure data transmission. CHILD SA is an IKEv2 term that is similar to IPsec SA in IKEv1. IKEv2 uses UDP ports 500 and 4500 (IPsec NAT Traversal).

Additional CHILD SA can be created to establish a new tunnel. This message exchange is called CREATE_CHILD_SA and can be used to negotiate new Diffie-Hellman group values and encryption/hash algorithm combinations.

IPsec tunnel peer authentication can be done using certificates based on a Public Key Infrastructure (PKI) or using the Extensible Authentication Protocol (EAP).

IPsec(IKEv1) VPN

When a VPN is created using **IPsec(IKEv1)**, a secure VPN tunnel is established using the IPsec group of protocols together with IKEv1.

In this case, UserGate DCFW can work as VPN client and VPN server.

GRE/IPsec VPN

GRE (RFC 2784) is a tunneling protocol that can encapsulate the packets of various types of protocols inside IP tunnels, creating a virtual point-to-point link over an IP

network. GRE is used for managing the process of transmitting multiprotocol and multicast IP traffic between two or more sites that can only communicate using the IP protocol. Note that GRE does not provide confidentiality and integrity for the transmitted data. To that end, IPsec protocols are used together with GRE.

When GRE and IPsec are used together, two types of connection can be created: IPsec over GRE and GRE over IPsec.

In case of an IPsec over GRE connection, encrypted traffic is transmitted over an unencrypted GRE tunnel, meaning that GRE encapsulation follows IPsec encapsulation. A downside of IPsec over GRE is that multicast and broadcast packets are not supported.

GRE over IPsec connections allow you to combine the advantages of GRE (multicast and broadcast support) and IPsec (encrypted traffic transmission). A GRE over IPsec connection encapsulates the traffic packets in GRE and then transmits them over an encrypted link (IPsec encapsulation).

To configure GRE over IPSec, follow these steps:

Parameter	Description
Step 1. Configure a site-to-site VPN connection.	For more details on configuring a site-to-site VPN connection, see the Site-to-Site VPN Connections section.
Step 2. Configure a GRE tunnel.	For more details on configuring a GRE tunnel interface, see the Tunnel Interface section. Important! When configuring a GRE tunnel interface, make sure to specify the addresses of the VPN interfaces as the source (local) and destination (remote) IP addresses.

Site-to-Site VPN Connections

A VPN connection that makes it possible to interconnect the local networks of remote offices is called a Site-to-Site VPN.

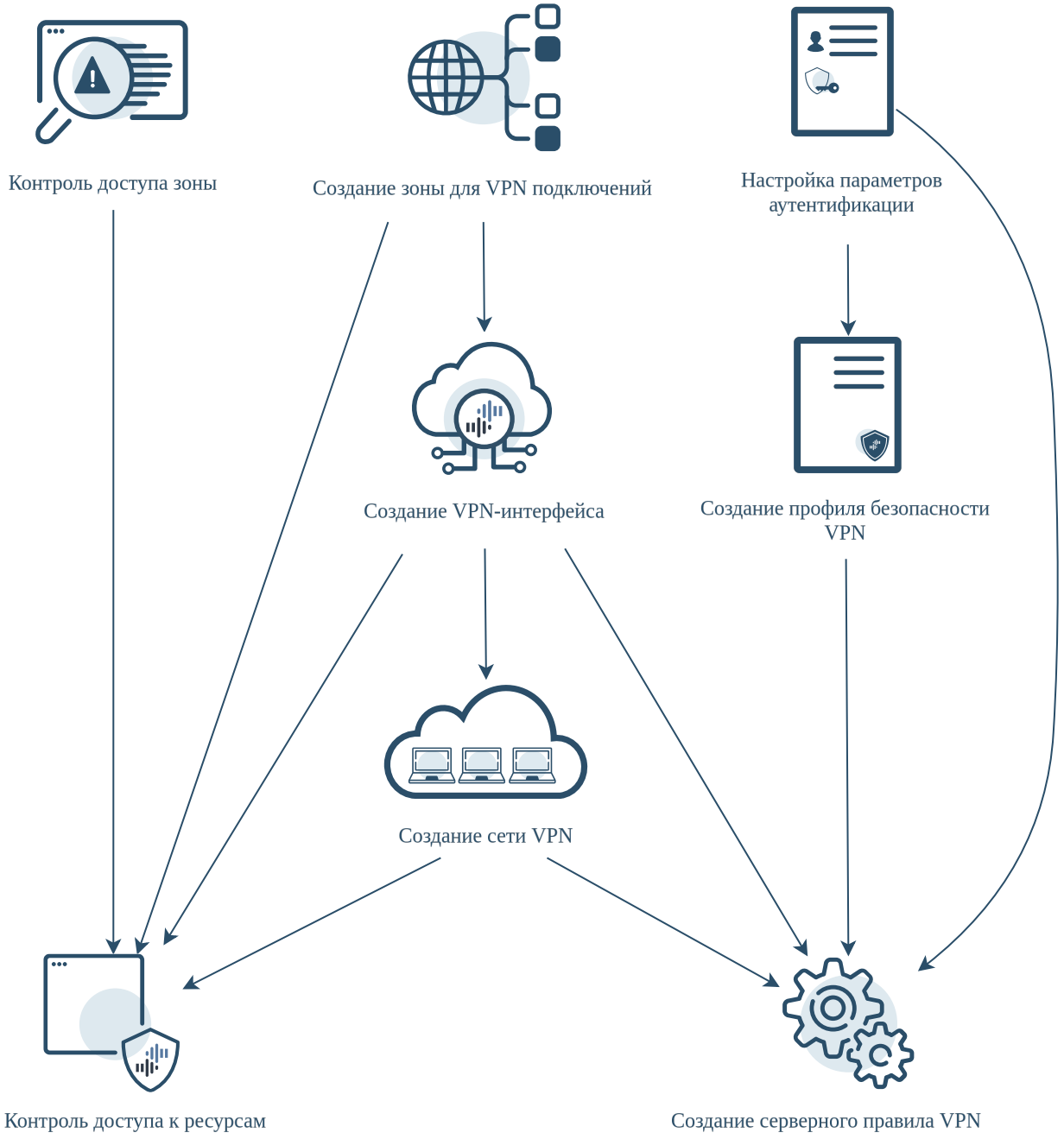
In this case, one firewall works as a VPN server and another as a VPN client. The client initiates a connection to the server. A Site-to-Site VPN can be created between two UserGate firewalls or between a UserGate firewall and a third-party device.

To create a Site-to-Site VPN, L2TP/IPsec(IKEv1), IPsec(IKEv2), IPsec(IKEv1) protocols are used.

Configure the relevant settings at both endpoints of the secure connection, i.e., the VPN server and VPN client.

Configuring VPN server

Configuring VPN server at DCFW includes the following main stages:



1. Zone access control.
2. Creating a zone for VPN connections.
3. Configuring authentication settings.

4. Creating a VPN security profile.
5. Creating a VPN interface.
6. Creating a VPN network.
7. Creating a VPN server rule.
8. Control of access to resources.

Zone Access Control

Allow the VPN service in the access control zone from which VPN clients will connect.

You can do it in the **Network → Zones** section of the Admin console. Then edit the access control settings for the zone from which VPN clients will connect and enable the VPN service. Usually, this is the **Untrusted** zone. For more information on creating and configuring zones, refer to the Zone Configuration section.

Creating a zone for VPN connections

Create a zone where the nodes connecting using a VPN will be placed.

To create a zone, use the **Network → Zones** section of the Admin console. This zone can later be used in security policies. For more information on creating and configuring zones, refer to the Zone Configuration section.

Configuring Authentication Settings

If the VPN tunnel is created using the **L2TP** protocol, you must create a local account on the VPN server. This account will be used to authenticate a node acting as VPN client. To create a local account, use the **Users and devices → Users** section. For convenience, all such users thus created can be placed in the existing **VPN servers** group that will be granted VPN connection access. For more details on creating user and group accounts, please read the Users and groups section of the guide.

If the **IPsec** secure connection is created, the following authentication methods for the remote node can be used:

- Authentication based on the **pre-shared key**. It is used when **IKEv1** protocol is used to create the secure connection. The pre-shared key is specified in the VPN security profiles. To establish a connection successfully, it should be identical on the VPN server and the VPN client.

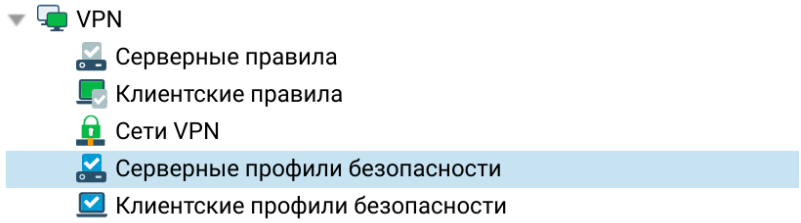
- Authentication based on the **certificates** using the Public Key Infrastructure (PKI). It is used when **IKEv2** protocol is used to create the secure connection. You need to create the client and server certificates in advance and import them to DCFW. For examples of how to create and use certificates for IKEv2 VPN, refer to the Appendix.

If you need to authenticate **VPN users**, you need to create the corresponding authentication profile. To create the authentication profiles, use the **Users and devices → Authentication profiles** section. The same authentication profile may be used that you use to authenticate users for Internet access. Note that transparent authentication methods such as Kerberos, NTLM, or SAML IDP cannot be used for VPN authentication. For more details on authentication profiles, see the Authentication Profiles section.

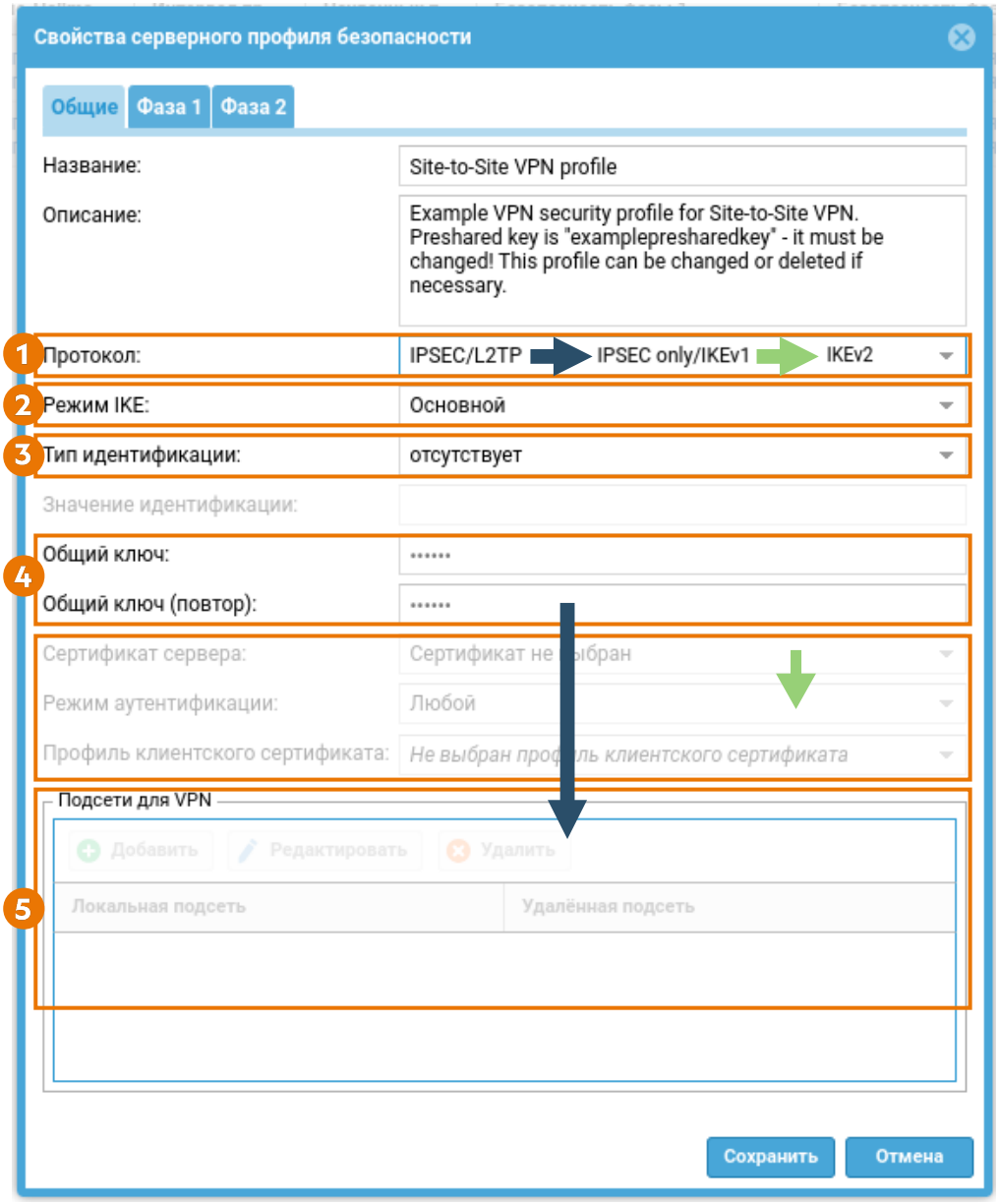
Creating a VPN Security Profile

In the VPN security profile settings, the types and settings of encryption and authentication algorithms are defined. Multiple security profiles may be used for connecting to different client types.

Security profiles for the VPN server and VPN client hosts are configured separately in the **VPN** section of the admin web console.



To create a **VPN server** security profile, go to **VPN → Server security profiles**, click **Add**, and fill in the required fields in the security profile properties:



In the **General** tab, you can select the VPN protocol version and set parameters for node authentication when establishing the secure connection.

1. **Protocol**. The options are as follows:

    - **IPsec/L2TP.**
    - **IPsec only/IKEV1**.
    - **IKEv2.**

2. **IKE mode** (for IKEv1 only). The options are as follows:

    - **Main**.
    - **Aggressive**.

3. **ID type** (the IKE local ID parameter). This is required for DCFW identification on a neighbor node when establishing a VPN connection to certain vendors' equipment. Enumerated selection options:

- **None**: the default value of the field. Used when the IKE local ID parameter is not required for establishing a VPN connection. For example, when a VPN connection between two UserGate nodes is established.
- **IPv4**: the host's IP address.
- **FQDN**: the host's address in the fully-qualified domain name (FQDN) format.
- **CIDR**: the host's address in the classless inter-domain routing (CIDR) format.
- **ID value**: the IKE local ID value in the format specified above.

4. Authentication type of the remote node when establishing the secure connection.

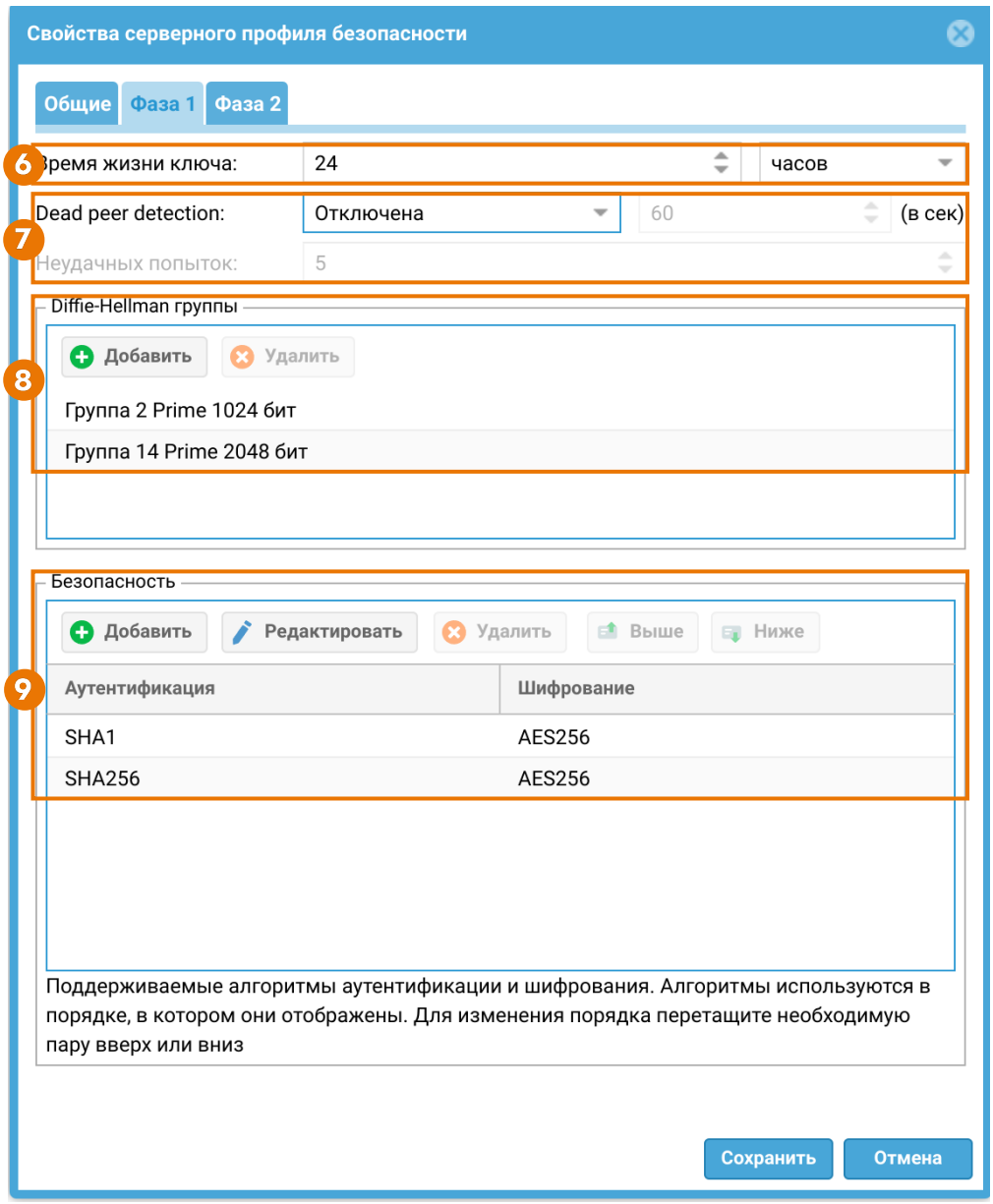
- If you choose IKEv1 protocol, you need to use authentication based on the **pre-shared key** . You need to specify the pre-shared key. This string must match on the VPN client and VPN server for a successful connection.
- If you choose IKEv2 protocol to establish the site-to-site tunnel, you can use authentication based on the certificates that use the Public Key Infrastructure (PKI). You need to specify the server certificate and client certificate profile created earlier. For examples of how to create and use certificates for IKEv2 VPN, refer to the Appendix. For more details about creating client certificates, see the Client Certificate Profiles section.

5. **Subnets for VPN**. They are specified if IPsec only/IKEV1 protocol is used to establish the VPN tunnel:

- **Local subnet: the IP address of an allowed local subnet.**
- Remote subnet: the IP address of an allowed subnet on the remote node side.

Next, the cryptographic parameters for the first and second phases of secure connection negotiation need to be configured.

In the first phase, an IKE SA is negotiated and established. Provide the following settings:



6. **Key lifetime**: the time period after which the parties re-authenticate and re-negotiate the first-phase settings.

7. **Dead peer detection** (DPD): to check that the channel is working and to disconnect/reconnect the channel if the connection is lost. DPD sends R-U-THERE messages periodically to check if the IPsec neighbor is available. There are 3 operating modes of the mechanism:

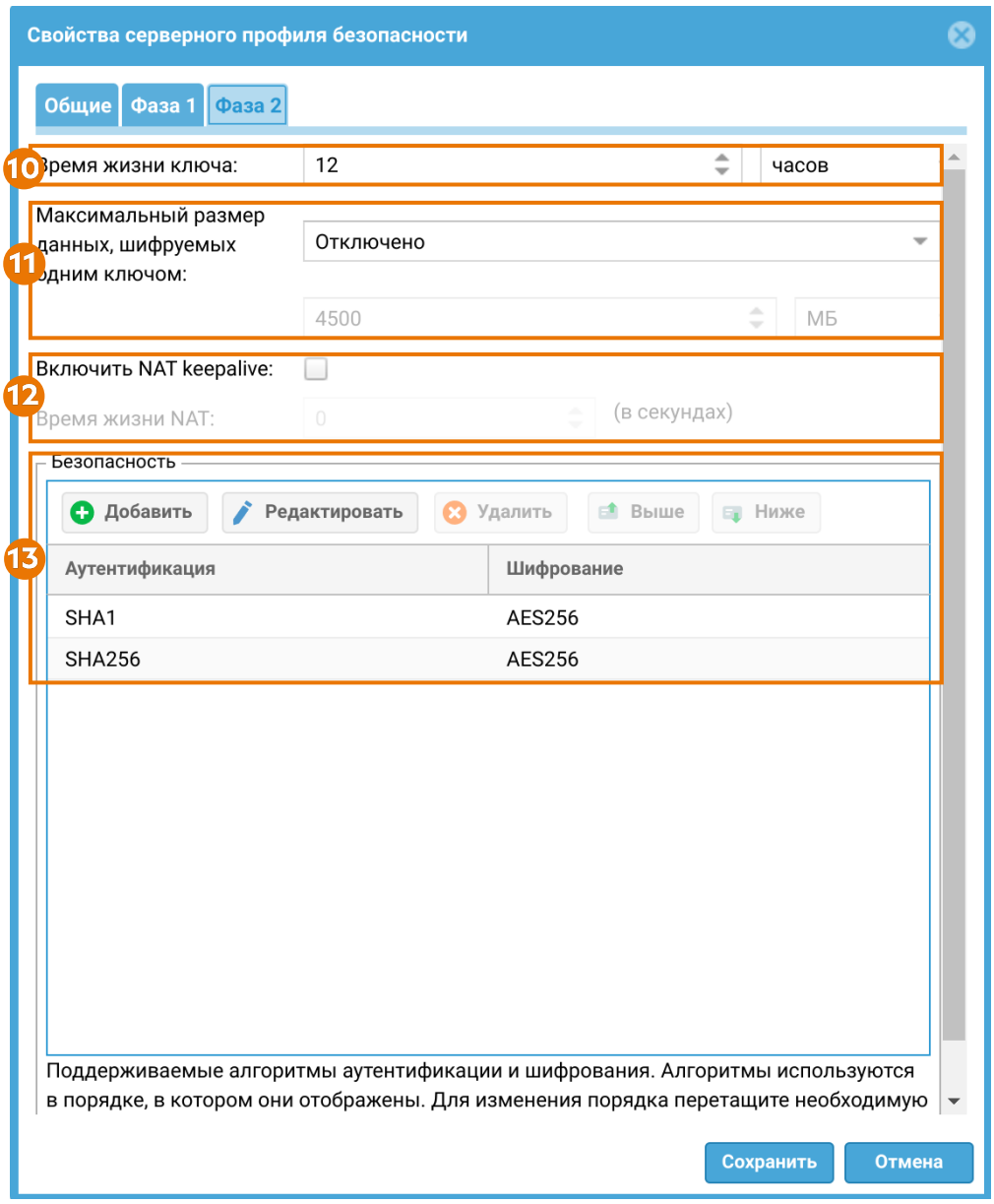
- **off**: the mechanism is disabled. DPD requests are not sent.
- **always on**: DPD requests are always sent within the specified time interval. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **Failures** field. If there is a response, the mechanism returns to the initial interval for sending DPD requests, and if there is no response, the connection is terminated.

- **Idle**: DPD requests are not sent while there is ESP traffic through the created SAs. If there are no packets within twice the specified time interval, then a DPD request is sent. If there is a response, a new DPD request will be sent again after a double interval of the specified time. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **Failures** field. If there is no response, the connection is terminated.

8. **Diffie-Hellman groups**: select the Diffie-Hellman groups that will be used for key exchange.

9. **Security**: select authentication and encryption algorithms. Algorithm are used in the order they are listed here. To reorder the algorithms, drag and drop them with the mouse or use the **Up**/**Down** buttons.

In the second phase, the method for securing data in the IPsec connections is selected. Provide the following settings:



10. **Key lifetime**: the time period after which the nodes must rotate the encryption key. The lifetime for the second phase is shorter than for the first one, which entails a more frequent key rotation.

11. **Key lifesize**: the key lifetime can also be expressed in bytes and is called lifesize in that case. If both values (**Key lifetime** and **Key lifesize**) are specified, the counter that reaches the limit first will trigger session key re-generation.

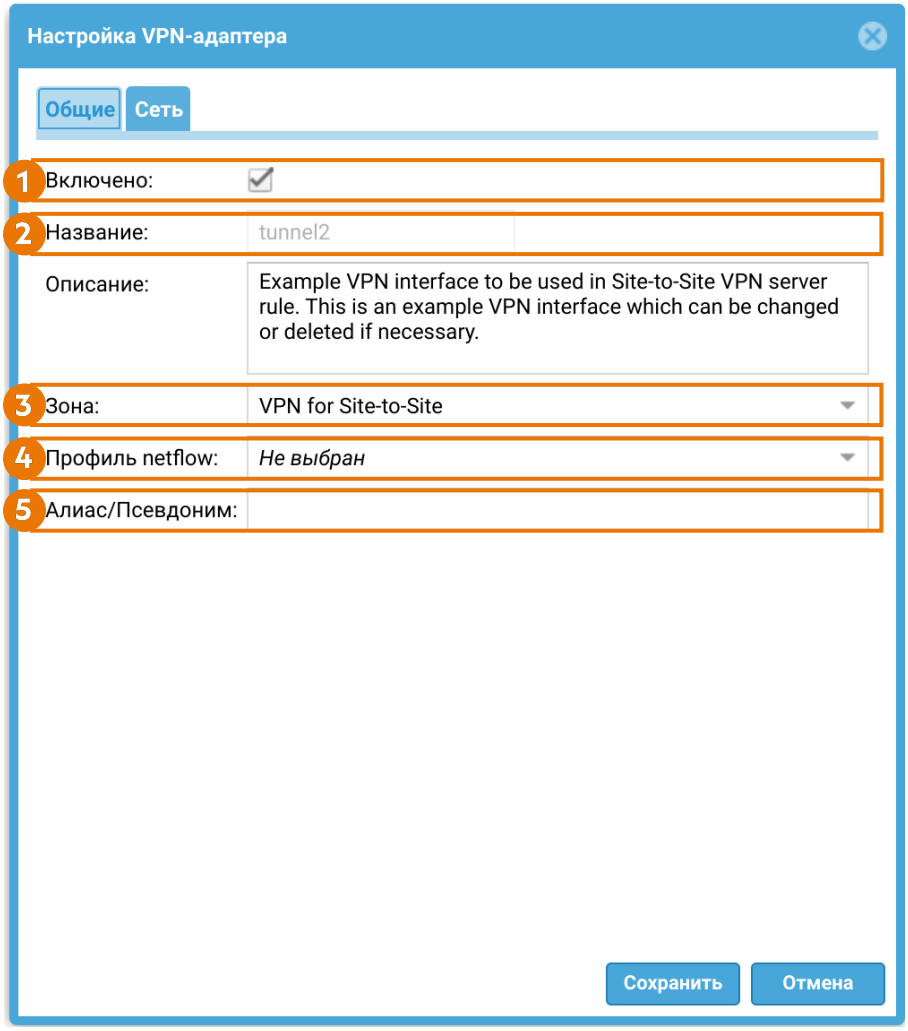
12. **NAT keepalive**: used in scenarios when IPsec traffic goes through a NAT node. NAT table entries are active for a limited time. If there was no VPN traffic over the tunnel during that time span, NAT table entries on the NAT host will be deleted, preventing further passage of VPN traffic. The VPN server located behind the NAT gateway uses NAT keepalive function to periodically send keepalive packets to a peer node in order to keep the NAT session active.

13. **Security**: the algorithms are used in their listing order. To reorder the algorithms, drag and drop them with the mouse or use the **Up**/**Down** buttons.

Creating a VPN Interface

A VPN interface is a virtual network adapter that will be used to connect VPN clients. This is a cluster-type interface, which means that it will be created automatically on all UserGate nodes included in a configuration cluster. If an HA cluster exists, in case any problems are identified with the active server, VPN clients will be automatically switched to a backup server, and without terminating existing VPN connections.

To create a VPN interface, use the **Network → Interfaces** section of the Admin console. Click the **Add** button, select **Add VPN** and specify the necessary parameters in the VPN device settings:



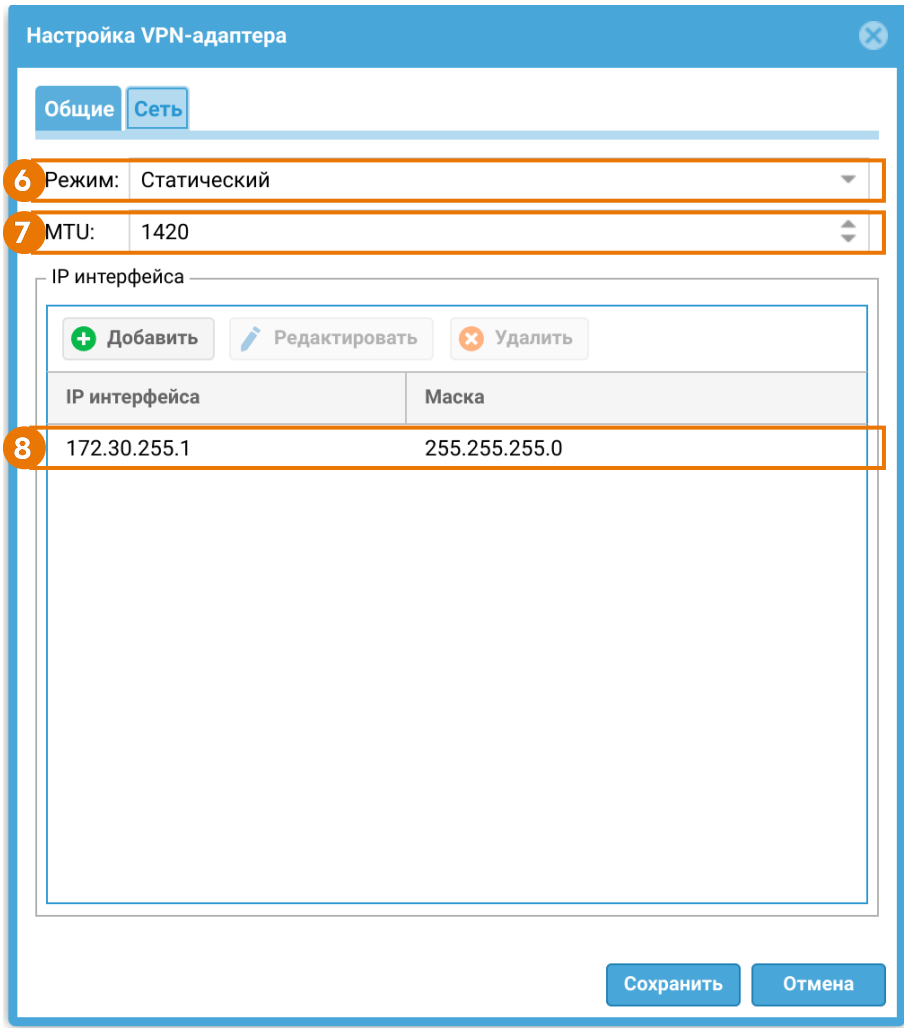
1. **Enabled**: enables or disables the interface.

2. **Name**: the name of the interface. Should be in the form *tunnelN*, where *N* is the ordinal number of the VPN interface.

3. **Zone**: the zone to which this interface will belong. All clients with a VPN connection to DCFW will be placed in this zone as well. In this field, you need to specify the zone created earlier at the stage of creating the zone for VPN connections.

4. **Netflow profile**: the Netflow profile used for this interface. For more details on Netflow profiles, see the Netflow Profiles section. (Optional)

5. **The interface's alias**. (Optional)



6. **Mode**: IP address assignment type. The options are no address, a static IP address, or a dynamic IP address obtained using DHCP. If the interface is to be used for receiving VPN connections (Site-2-Site VPN or Remote access VPN), a static IP address must be used.

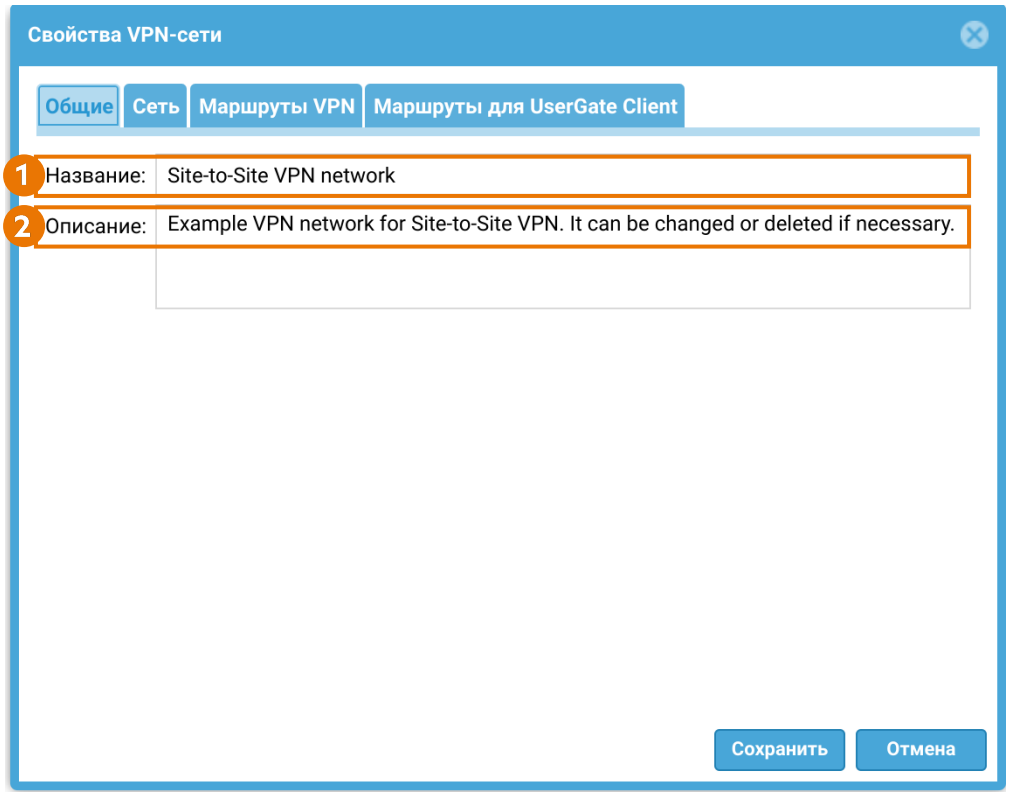
7. **MTU**: the MTU size for the selected interface. If packets transmitted over the VPN tunnel exceed the maximum MTU at any of the intermediate devices, they can be split into fragments. This can increase the latency and reduce the performance. By setting an optimum MTU value on the tunnel interface, you can avoid packet fragmentation and reduce the latency.

8. This field is used to specify the **IP address** of the VPN interface if a static IP address is used.

Creating a VPN Network

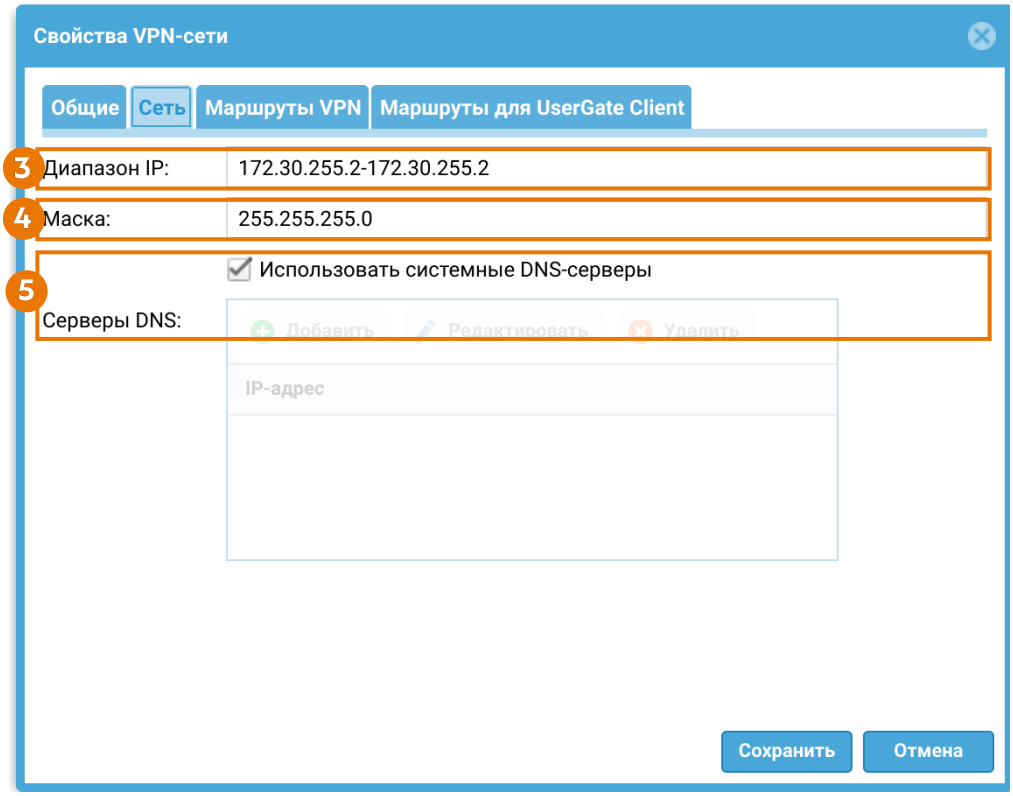
A VPN network determines the network settings that will be used for connecting the client to the server. This is primarily the assignment of IP addresses to the clients inside the tunnel, the DNS settings, and the routes that will be passed to the clients that support the use of routes assigned to them. Multiple tunnels may be used with different settings for different clients.

To create the VPN network, use the **VPN → VPN networks** section of the Admin console. You need to click **Add** and fill the necessary parameters in the VPN network properties:



1. **Name** of the VPN network.

2. **Description** of the VPN network. (Optional)

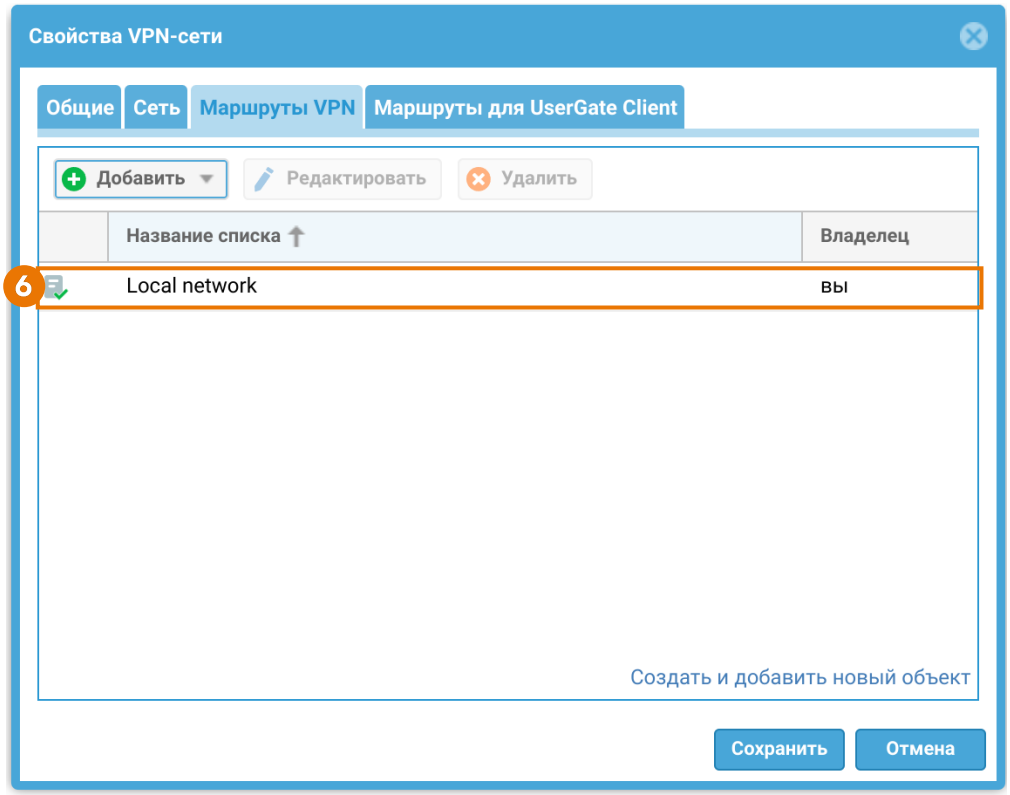


3. **IP addresses range**, which will be used by the clients. You must exclude the address assigned to the **VPN interface** of DCFW used along with this network from this range. Do not enter network addresses or the broadcast address here.

4. **Mask** of the VPN network.

5. Specify the **DNS servers** that will be passed to the client or set the **Use system DNS** checkbox, in which case the client will be assigned the DNS servers used by DCFW.

> **Important!**
>
> **A maximum of two DNS servers can be specified.**

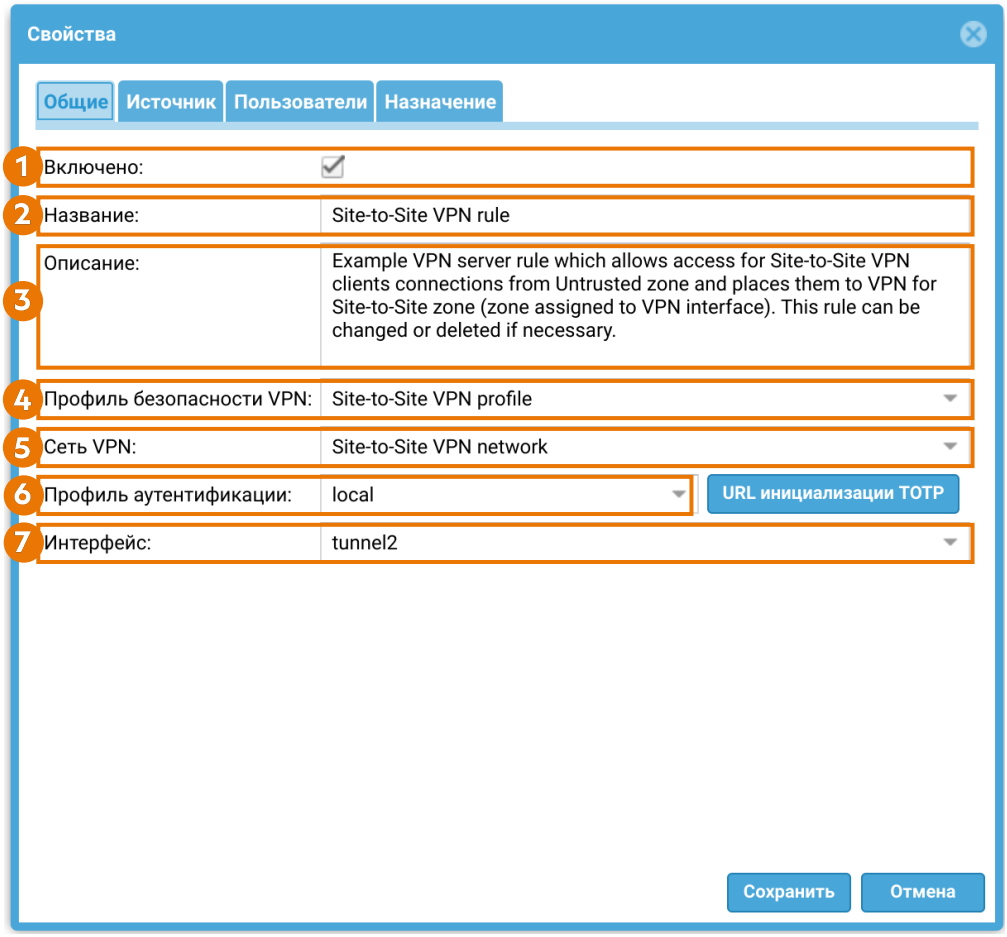


6. **VPN routes**: the routes sent to the VPN client in the CIDR format or a predefined IP address list.

The **UserGate Client routes** tab is not used for configuring Site-to-Site VPN connections. It is used to configure split tunneling for UserGate Client for remote access to the network.

Creating a VPN Server Rule

To create VPN server rules, use the **VPN → Server rules** section of the Admin console. Then click **Add** and fill in the relevant fields in the rule properties:



1. **Enabled**: enables or disables the VPN rule.

2. VPN server rule **name**.

3. VPN server rule **description**. (Optional)

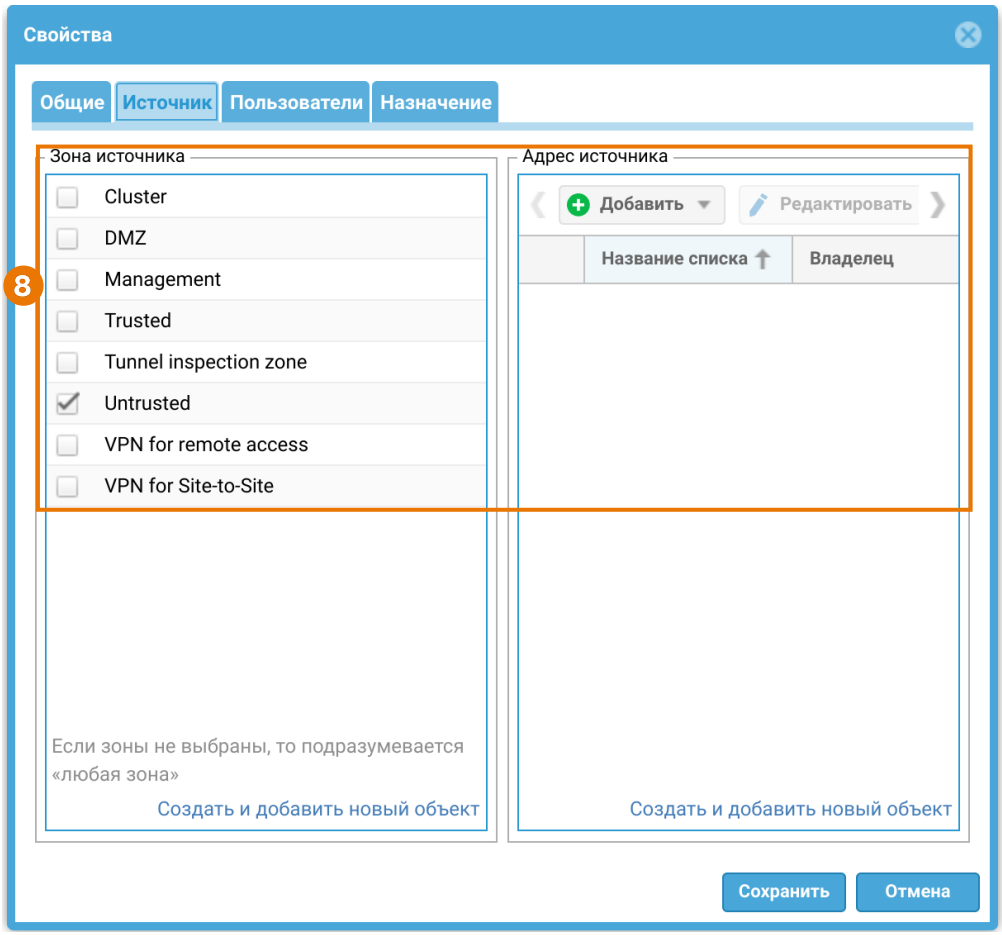
4. **VPN security profile**: the security profile created earlier.

5. **VPN network**: the network created earlier, at the stage of creating VPN network. If you configure the server rule for the IPsec tunnel and the UserGate acts as the VPN server (the VPN protocol **IPsec only/IKEV1** is specified in the server security profile), choose **Do not use** in this field.

6. **Authentication profile**: authentication profile for VPN users. The same authentication profile may be used that you use to authenticate users for Internet access. Note that transparent authentication methods such as Kerberos, NTLM or SAML IDP cannot be used for VPN authentication. If necessary, you can create an authentication profile for VPN users in the **Users and devices → Authentication profiles** section. For more details on authentication profiles, see the Authentication profiles section.

If you configure the server rule for the IPsec tunnel and the UserGate acts as the VPN server (the VPN protocol **IPsec only/IKEV1** is specified in the server security profile), choose **Do not use** in this field.

7. **Interface**: the VPN interface created earlier.

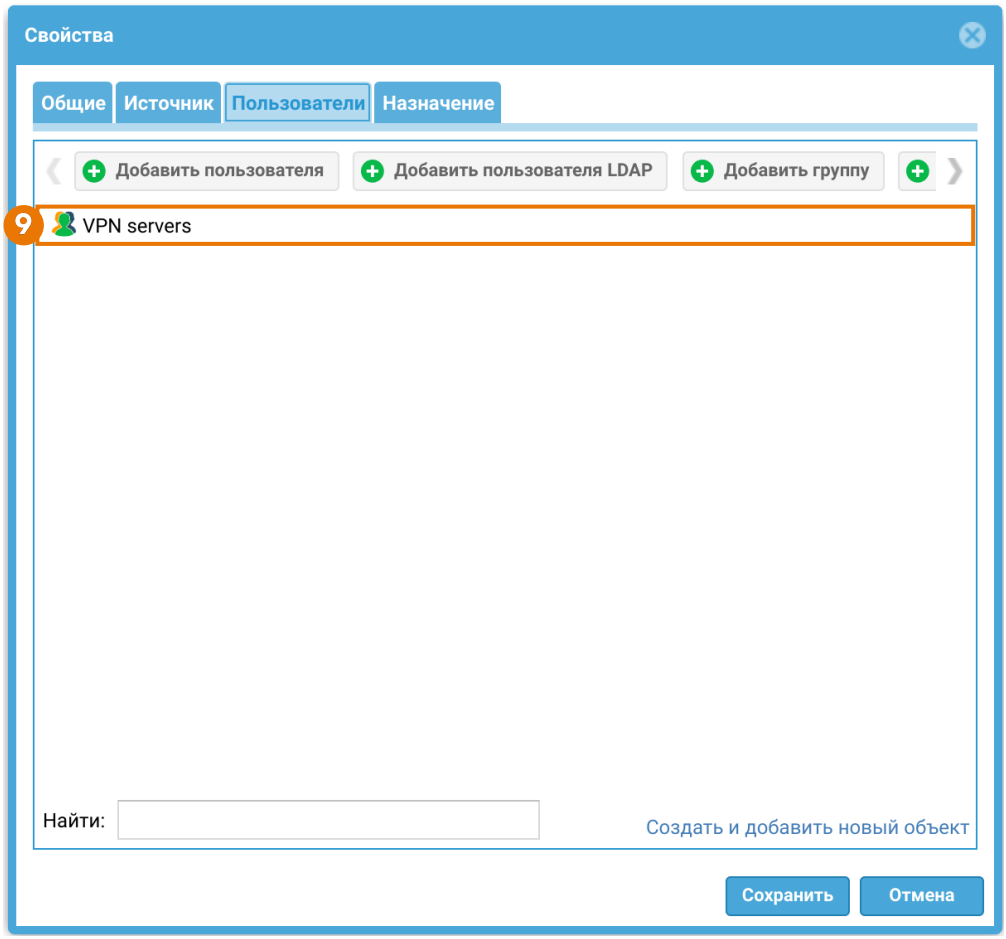


8. **Source**: the zones and IP addresses from which VPN connections are allowed. Normally, the clients are on the Internet, so specify the **Untrusted** zone.

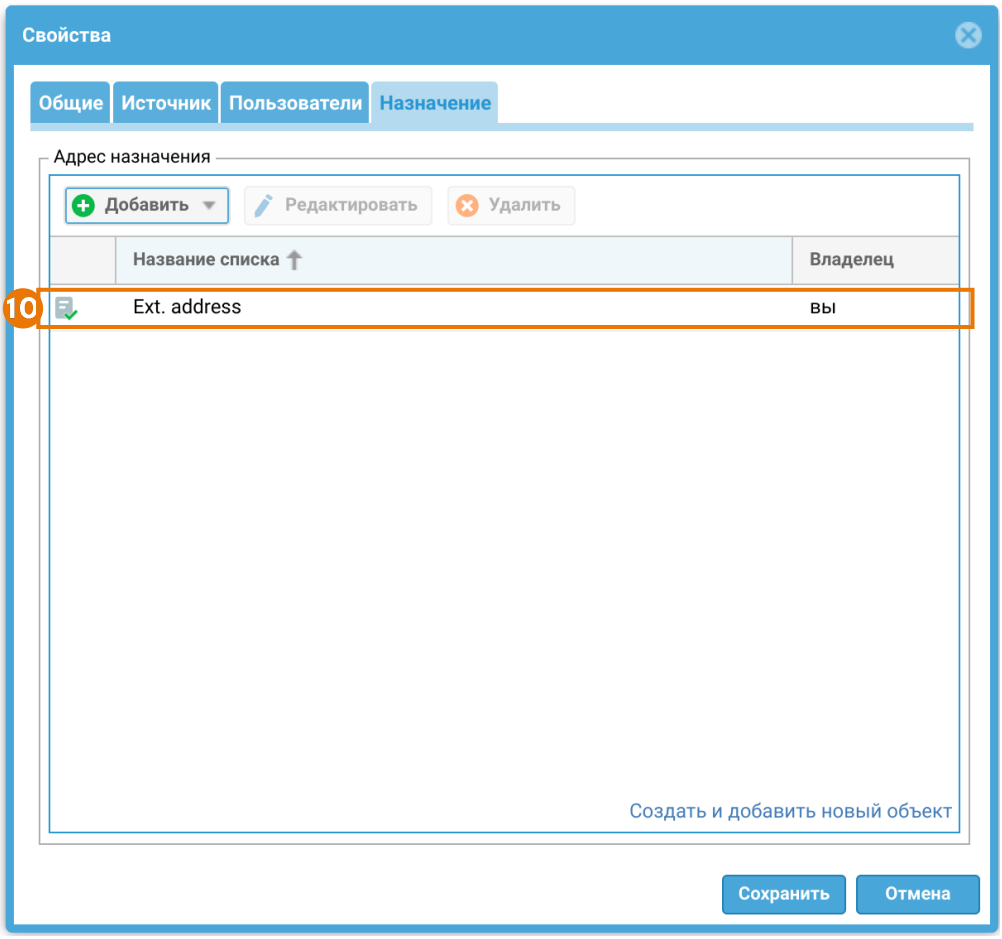
Important!

The traffic processing logic is as follows:

- **The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.**
- **The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.**



9. **Users**: a group of server accounts or individual server accounts for which VPN connections are allowed.



10 **Destination**: one or more interface addresses to which the clients will connect. The interface must belong to the zone specified at the stage of zone access control.

Important!

To apply different server rules to different clients, use the *Source zone* and *Source address* settings. The *Users* setting does not govern the selection of a server rule, as the user is checked only after the VPN connection has been established.

Note

When changing the VPN server settings (changing server rules, changing security profiles, adding new VPN networks), the VPN server does not reboot, so previously established active VPN client sessions are not terminated. A reboot of the VPN server and reconnection of active VPN client sessions may occur if the IP address of the tunnel interface of the VPN server is changed.

> **Note**
> **Starting from software version 7.2.0, the traffic transferred over a VPN connection between the offices is marked in logs with the logins of specific users if they are known, and with the word "Unknown" if they are not known.**

Control of Access to Resources

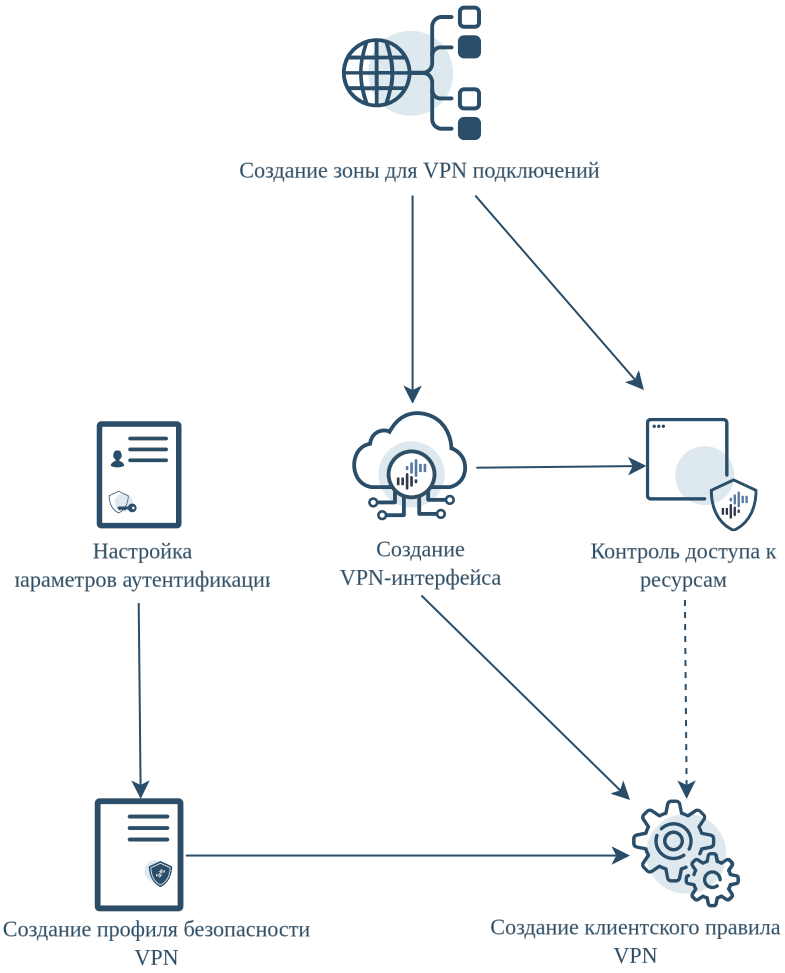
To grant VPN users access to certain network segments or, for example, Internet, go to **Network policies → Firewall** and create a firewall rule that allows traffic from the zone for VPN connections to the desired zones. For more details on configuring firewall rules, see the Firewall section of the guide.

To let the traffic be passed back to the client from the allowed zones via the VPN tunnel, you need to create an "allow" firewall rule and specify the desired source zone and destination zone, for example, the VPN connection zone, which was configured earlier.

You need to configure routing for returning traffic on the VPN server. For example, to provide the information on client subnets to the VPN server, you need to configure the static route in the virtual router properties (**Network → Virtual routers**) and specify the VPN tunnel address used on the VPN client as the destination address. For details on virtual router parameters, see the Virtual Routers section.

Configuring VPN Client

Configuring VPN client at DCFW includes the following main stages:



1. Creating a zone for VPN connections.
2. Creating a VPN interface.
3. Control of access to resources.
4. Configuring authentication settings.
5. Creating a VPN security profile.
6. Creating a VPN client rule.

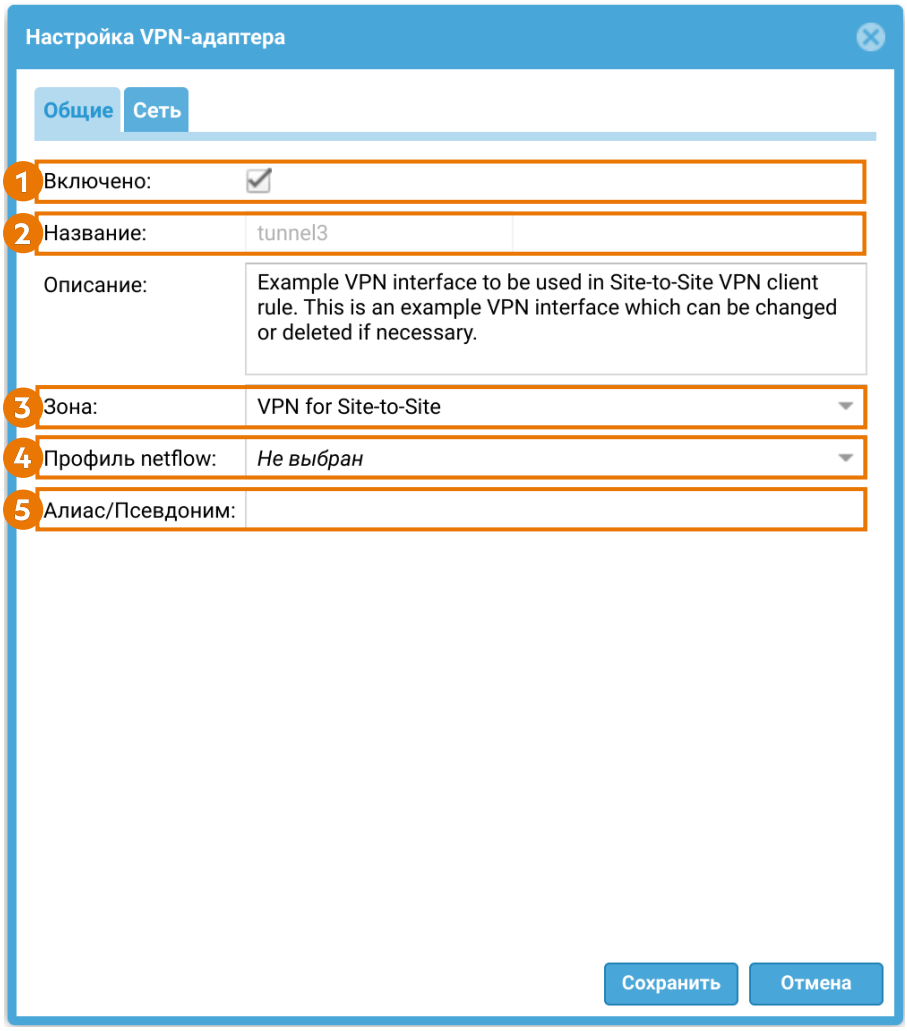
Creating a zone for VPN connections

Create a zone where the interfaces used for VPN connections will be placed. To create a zone, use the **Network → Zones** section of the Admin console. For more information on creating and configuring zones, refer to the Zone Configuration section.

Creating a VPN Interface

A VPN interface is a virtual network adapter that will be used for VPN connection. This is a cluster-type interface, which means that it will be created automatically on all UserGate nodes included in a configuration cluster. If an HA cluster exists, in case any problems are identified with the active server, VPN clients will be automatically switched to a backup server, and without terminating existing VPN connections.

To create a VPN interface, use the **Network → Interfaces** section of the Admin console. Click the **Add** button, select **Add VPN** and specify the necessary parameters in the VPN device settings:



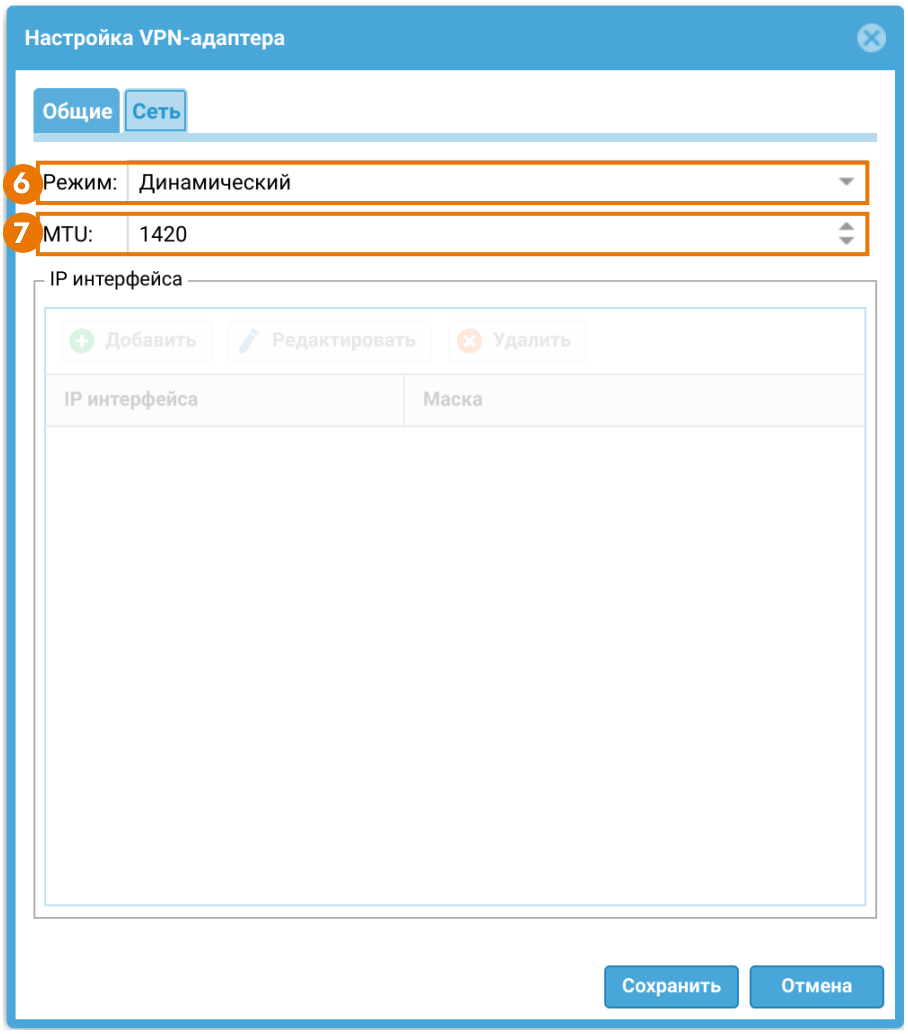
1. **Enabled**: enables or disables the interface.

2. **Name**: the name of the interface. Should be in the form *tunnelN*, where *N* is the ordinal number of the VPN interface. The **Description** field below is optional.

3. **Zone**: the zone to which this interface will belong. In this field, you need to specify the zone created earlier at the stage of creating the zone for VPN connections.

4. **Netflow profile**: the Netflow profile used for this interface. For more details on Netflow profiles, see the Netflow Profiles section. (Optional)

5. **The interface's alias**. (Optional)



6. **Mode**: IP address assignment type. The options are as follows: no address, static IP address or dynamic IP address obtained via DHCP. To use the interface as a client VPN interface, use the **Dynamic** IP assignment mode. When the connection is established, the interface will be assigned an IP address from the VPN network address range configured in the VPN server properties at the stage of creating VPN network.

7. **MTU**: the MTU size for the selected interface. If packets transmitted over the VPN tunnel exceed the maximum MTU at any of the intermediate devices, they can be split into fragments. This can increase the latency and reduce the performance. By setting an optimum MTU value on the tunnel interface, you can avoid packet fragmentation and reduce the latency.

Important!

If you select the same example tunnel interface with the default settings in the VPN server and VPN client configuration sections, an IP address conflict will arise during the establishment of a client-to-server connection. For things to work correctly, the address ranges of the tunnel interfaces should not overlap. Make sure to set unique address ranges on the client and server.

Control of Access to Resources

If necessary, create an "allow" firewall rule, which allows traffic between the VPN connection zone and destination zones, in the **Network policies → Firewall** section.

To let the traffic pass to the server via the VPN tunnel from the desired client server zone, you need to create an "allow" firewall rule, specifying the desired source zone and destination zone, for example, the VPN connection zone**.** For more details on how to create and configure firewall rules, see the Firewall section of the manual.

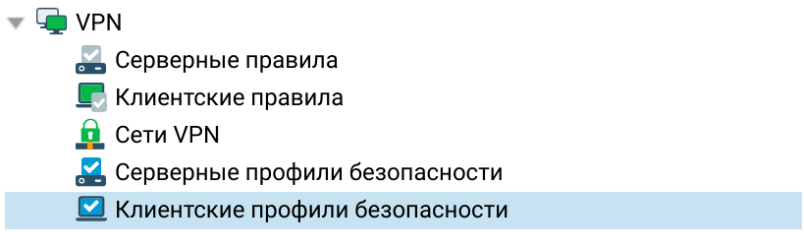
Configuring authentication settings

When the **IPsec** secure connection using **IKEv2** protocol is established, authentication based on **certificates** using the Public Key Infrastructure (PKI) is used. You need to import the VPN client certificate created earlier in the **UserGate → Certificates** section on the device acting as the VPN client.

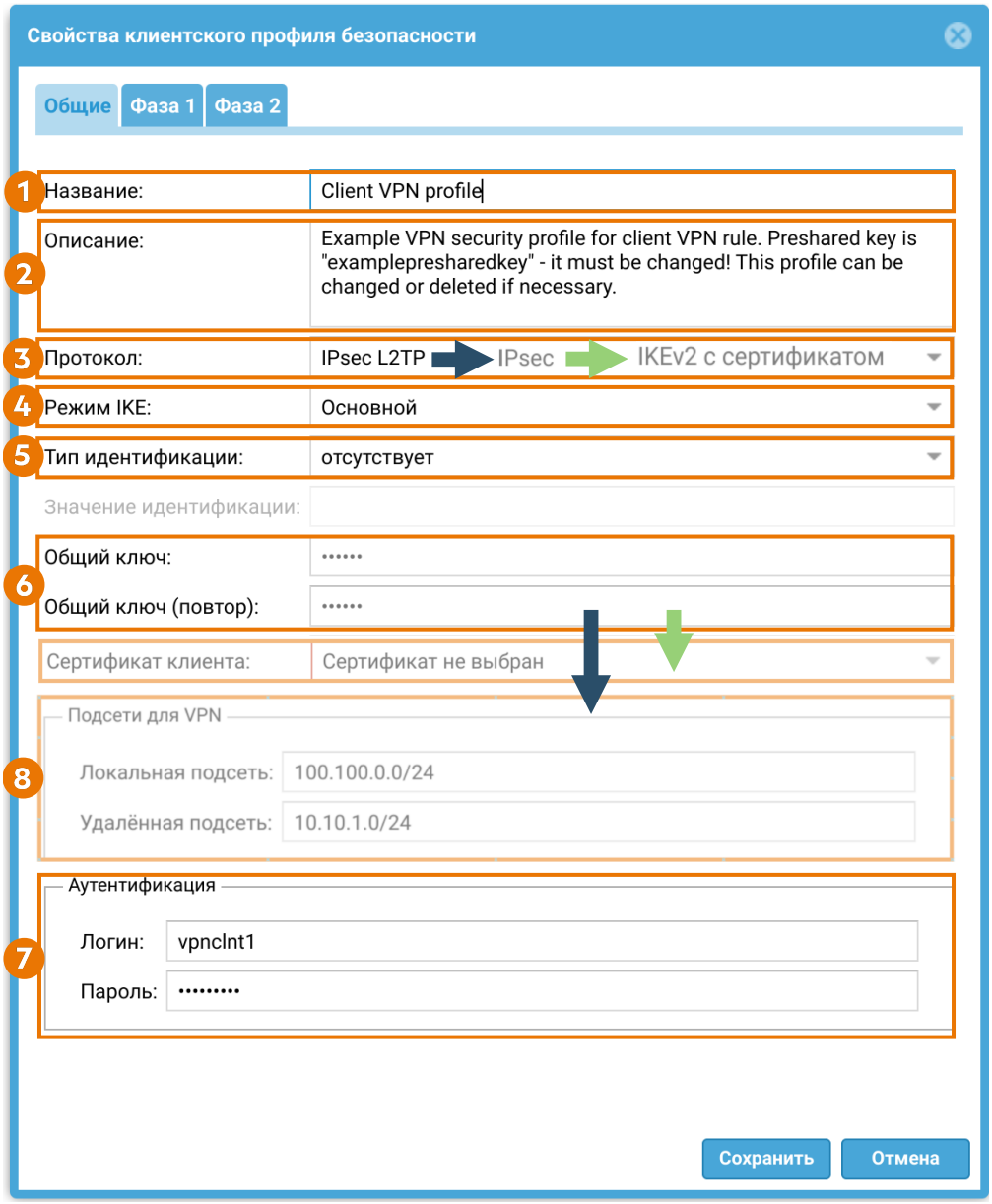
For examples of how to create and use certificates for IKEv2 VPN, refer to the Appendix.

Creating a VPN Security Profile

In the VPN security profile settings, the types and settings of encryption and authentication algorithms are defined. Security profiles for the VPN server and VPN client hosts are configured separately in the **VPN** section of the admin console:



To create a **VPN client** security profile, go to **VPN → Client security profiles**, click **Add**, and fill in the required fields in the client security profile properties:



In the **General** tab, you can select the IKE protocol version and set parameters for node authentication when establishing the secure connection.

1. **Name** of the client security profile.

2. **Description** of the client security profile. This parameter is optional.

3. **Protocol**: the protocol used to establish a VPN link between the two networks. The options are as follows:

- **IPsec L2TP**: create a secure VPN link using L2TP and IPsec/IKEv1
- **IPsec**: create a secure VPN link to a VPN server using IPsec/IKEv1
- **IKEv2 with certificate**: create a secure VPN link using IKEv2 with authentication using a certificate based on the Public Key Infrastructure (PKI).

4. **IKE mode**. The options are as follows:

- **Main**.
- **Aggressive**.

5. **ID type** (the IKE local ID parameter). This is required for the identification of the neighbor host when establishing a VPN connection to certain vendors' equipment. Enumerated selection options:

- **None**: the default value of the field. Used when the IKE local ID parameter is not required for establishing a VPN connection. For example, when a VPN connection between two UserGate nodes is established.
- **IPv4**: the host's IP address.
- **FQDN**: the host's address in the fully-qualified domain name (FQDN) format.
- **CIDR**: the host's address in the classless inter-domain routing (CIDR) format.
- **ID value**: the IKE local ID value in the format specified above.

6. Authentication type of the remote node when establishing the secure connection.

- If you choose **IPsec/L2TP** or **IPsec** protocol, you need to use authentication based on the pre-shared key. You need to specify the pre-shared key. This string must match on the VPN client and VPN server for a successful connection.
- If you choose **IKEv2 with a certificate** protocol to establish the site-to-site tunnel, authentication based on the certificates that use the Public Key Infrastructure (PKI) is used. You need to specify the client certificate created earlier. For examples of how to create and use certificates for IKEv2 VPN, refer to the Appendix.

7. **Authentication**: login and password of the local account created on the VPN server to authenticate the node acting as the VPN client when establishing the **L2TP** tunnel.

8. **Subnets for VPN**:

- **Local subnet: the IP address of an allowed local subnet.**
- **Remote subnet**: the IP address of an allowed subnet on the remote VPN server side.

Next, the cryptographic parameters for the first and second phases of secure connection negotiation need to be configured.

In the first phase, an IKE SA is negotiated and established. Provide the following settings:

9. **Key lifetime**: the time period after which the parties re-authenticate and re-negotiate the first-phase settings.

10. **Dead peer detection** (DPD): to check that the channel is working and to disconnect/reconnect the channel if the connection is lost. DPD sends R-U-THERE messages periodically to check if the IPsec neighbor is available. There are 3 operating modes of the mechanism:

- **off**: the mechanism is disabled. DPD requests are not sent.

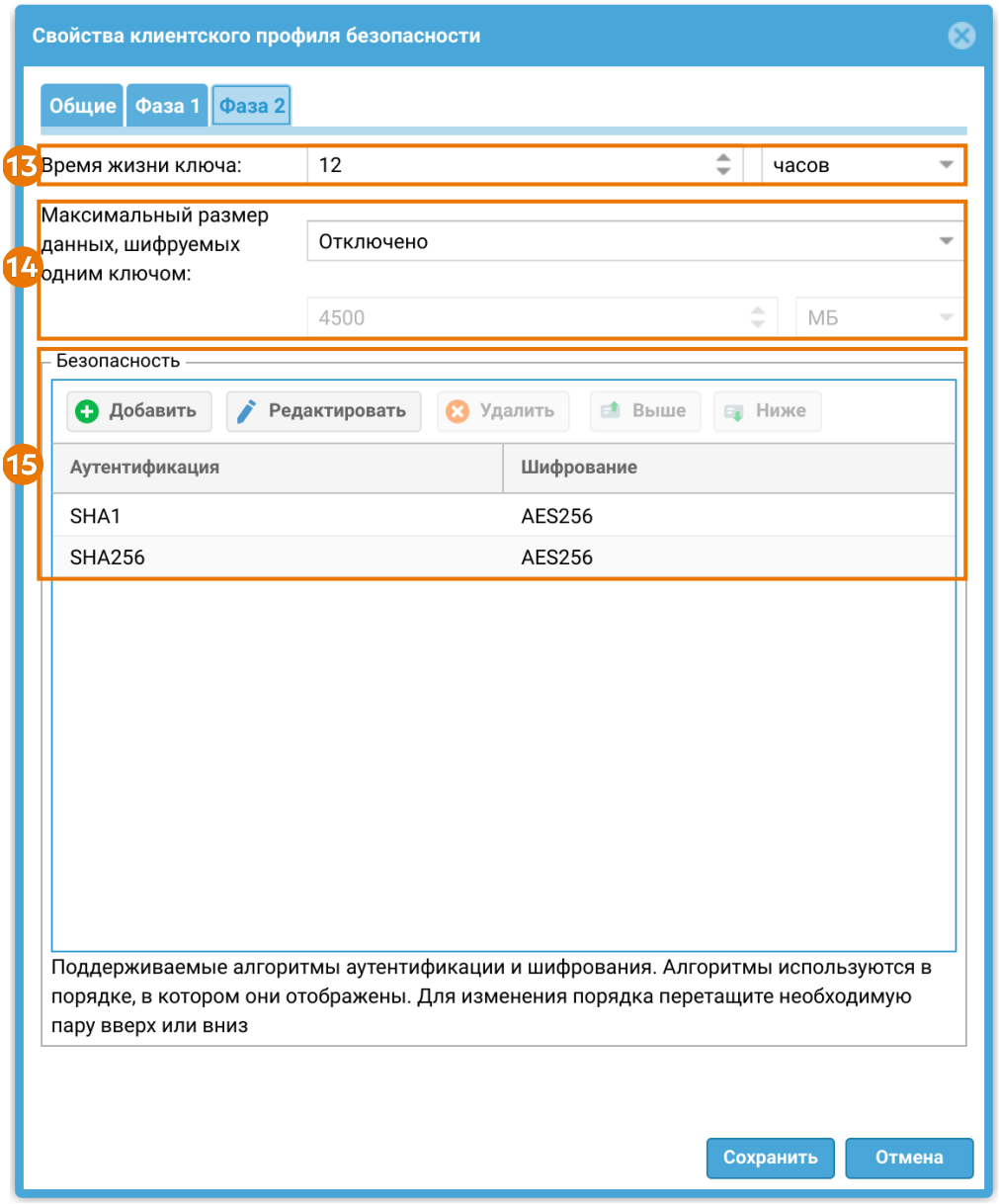
- **always on**: DPD requests are always sent within the specified time interval. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **Failures** field. If there is a response, the mechanism returns to the initial interval for sending DPD requests, and if there is no response, the connection is terminated.

- **Idle**: DPD requests are not sent while there is ESP traffic through the created SAs. If there are no packets within twice the specified time interval, then a DPD request is sent. If there is a response, a new DPD request will be sent again after a double interval of the specified time. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **Failures** field. If there is no response, the connection is terminated.

11. **Diffie-Hellman groups**: select the Diffie-Hellman groups that will be used for key exchange.

12. **Security**: here you can select authentication and encryption algorithms. To reorder the algorithms, drag and drop them with a mouse or use the **Up**/**Down** buttons.

In the second phase, the method for securing data in the IPsec connections is selected. Provide the following settings:



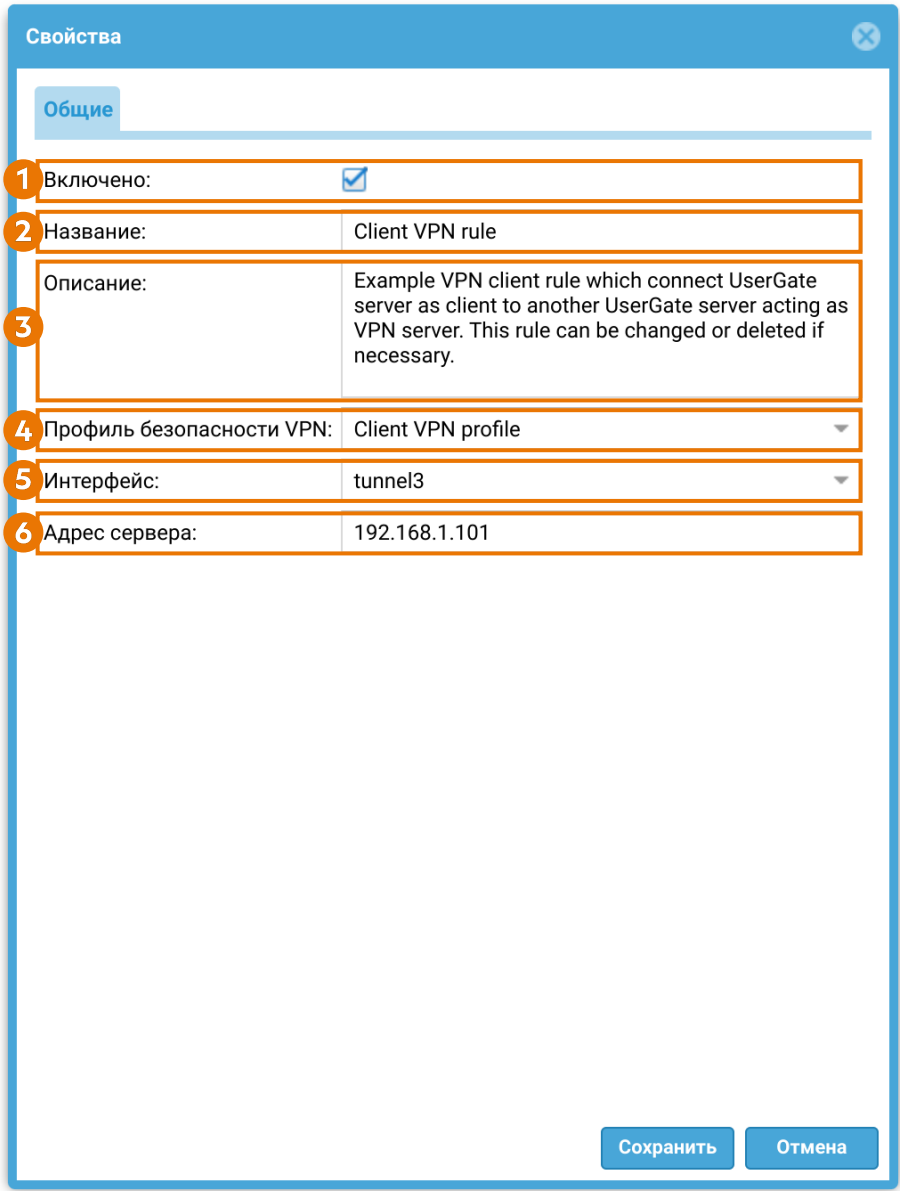
13. **Key lifetime**: the time period after which the nodes must rotate the encryption key. The lifetime for the second phase is shorter than for the first one, which entails a more frequent key rotation.

14. **Key lifesize**: the key lifetime can also be expressed in bytes and is called lifesize in that case. If both values (**Key lifetime** and **Key lifesize**) are specified, the counter that reaches the limit first will trigger session key re-generation.

15. **Security**: here you can select authentication and encryption algorithms. To reorder the algorithms, drag and drop them with a mouse or use the **Up**/**Down** buttons.

Creating a VPN client rule

The VPN client rule will initiate a VPN server connection. To create VPN client rules, use the **VPN → Client rules** section of the Admin console. Then click **Add** and fill in the relevant fields in the rule properties:



1. **Enabled**: enables or disables the rule.

2. **Name** of the client rule.

3. **Description** of the client rule (optional).

4. **Security profile**: the VPN client security profile created earlier.

5. **Interface**: the VPN interface created earlier.

6. **VPN server address**: the VPN server address (IP address, FQDN) to which this VPN client will connect.

When the VPN server and client have been configured, the client initiates a connection to the server, and if the settings are correct, a VPN tunnel is brought up. To bring down the tunnel, disable the VPN client rule (set on the client) or the VPN server rule (set on the server).

Remote Access VPN

A VPN connection that allows users to securely access a company's corporate network over the internet is called a Remote Access VPN.

In this connection, the UserGate firewall (DCFW) acts as a VPN server, and user devices act as VPN clients. UserGate Client can be used as client software on user devices, while native clients of most popular operating systems are also supported.

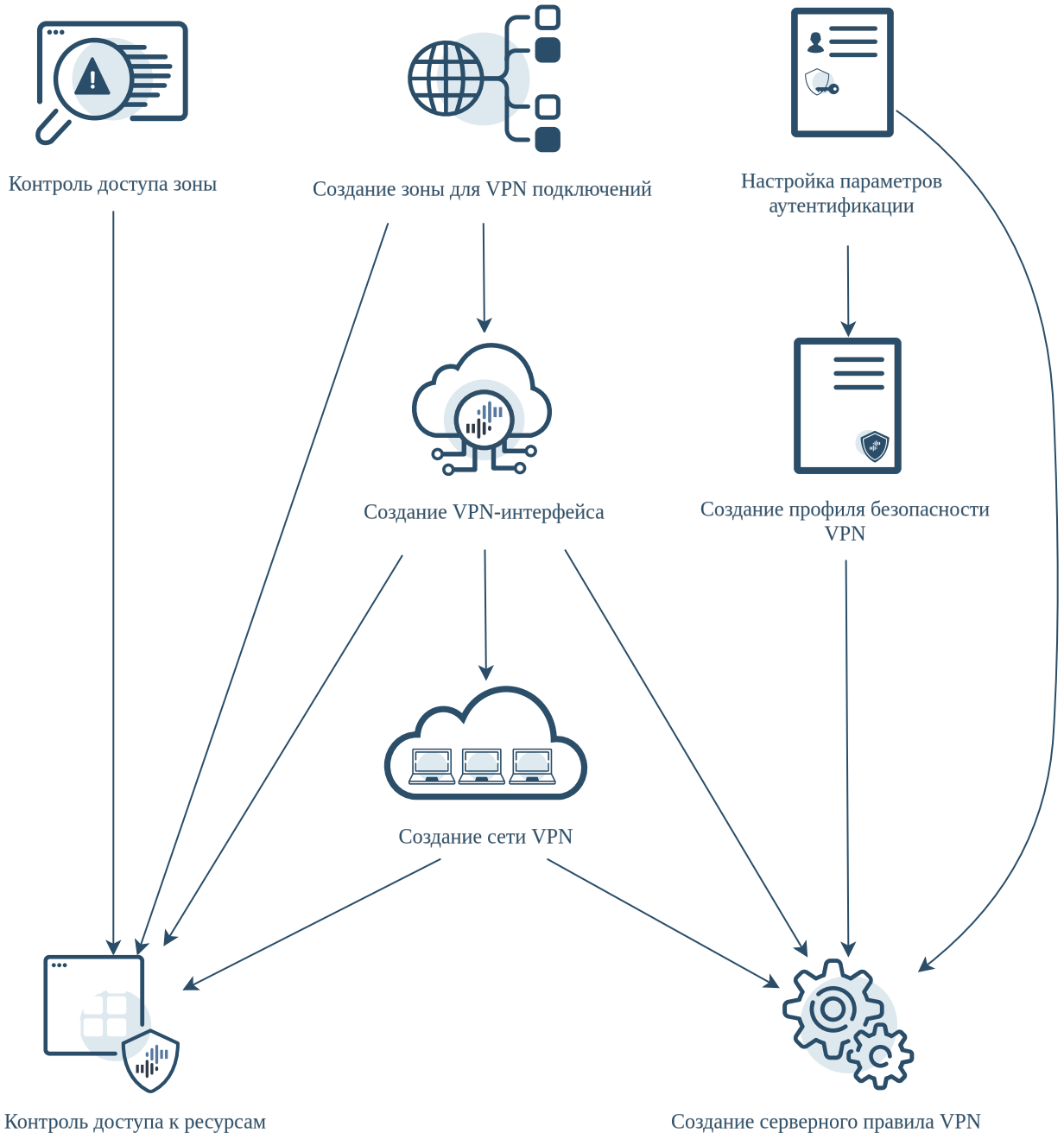
The following protocols are used to establish a secure connection:

- L2TP/IPSec(IKEv1);
- IPSec(IKEv2).

To create a Remote Access VPN, you need to configure the relevant settings on the VPN server, and then configure and connect a VPN client on the user's equipment.

Configuring VPN Servers

The algorithm of setting a VPN server on a UserGate DCFW includes the following key steps:



1. Zone access control.
2. Creating a zone for VPN connections.
3. Configuring authentication settings.
4. Creating a VPN security profile.
5. Creating a VPN interface.
6. Creating a VPN network.
7. Creating a VPN server rule.

8. Control of access to resources.

Zone Access Control

Allow the VPN service in the access control zone from which VPN clients will connect.

You can do it in the **Network → Zones** section of the Admin console. Then edit the access control settings for the zone from which VPN clients will connect and enable the VPN service. Usually, this is the **Untrusted** zone. For more information on creating and configuring zones, refer to the Zone Configuration section.

Creating a zone for VPN connections

Create a zone where the clients connected via VPN will be placed.

To create a zone, use the **Network → Zones** section of the Admin console. This zone can later be used in security policies. For more information on creating and configuring zones, refer to the Zone Configuration section.

Configuring Authentication Settings

1. If the **IPsec** secure connection is created, the following **node authentication** methods can be used:

- Authentication based on the **pre-shared key**. It is used when **IKEv1** protocol is used to create the secure connection. The pre-shared key is specified in the VPN security profile. To establish a successful connection, it should be identical on the VPN server and VPN clients.
- Authentication based on the **certificates** using the Public Key Infrastructure (PKI). It is used when **IKEv2** protocol is used to create the secure connection. You need to create the client and server certificates in advance, and import them to DCFW and client computers. Additionally, you have to create client certificate profiles on your DCFW. For examples of how to create and use certificates for IKEv2 VPN, refer to the Appendix. For information on how to create them, please see the Client certificate profiles section.

2. 2. To create the authentication profiles, use the **Users and devices → Authentication profiles** section. The same authentication profile may be used that you use to authenticate users for Internet access. Note that transparent authentication methods such as Kerberos, NTLM, or SAML IDP cannot be used for VPN authentication.

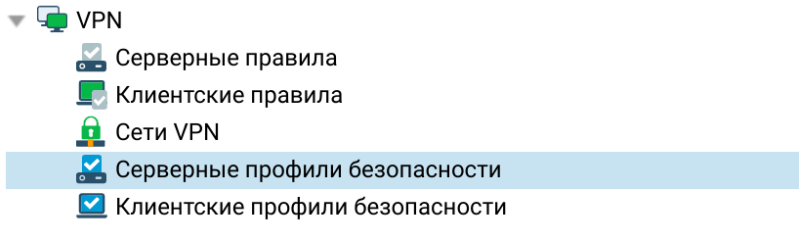
Multi-factor authentication can be used when authenticating VPN users. The second factor can be received in the form of TOTP single-use codes. VPN with TOTP works for UserGate Client only with IKEv2 (the code is entered in a separate window), for other clients — only with IKEv1 (the code is entered in the password separated by a colon: *user_password:totp_code*).

For more details on authentication profiles, see the Authentication Profiles section.

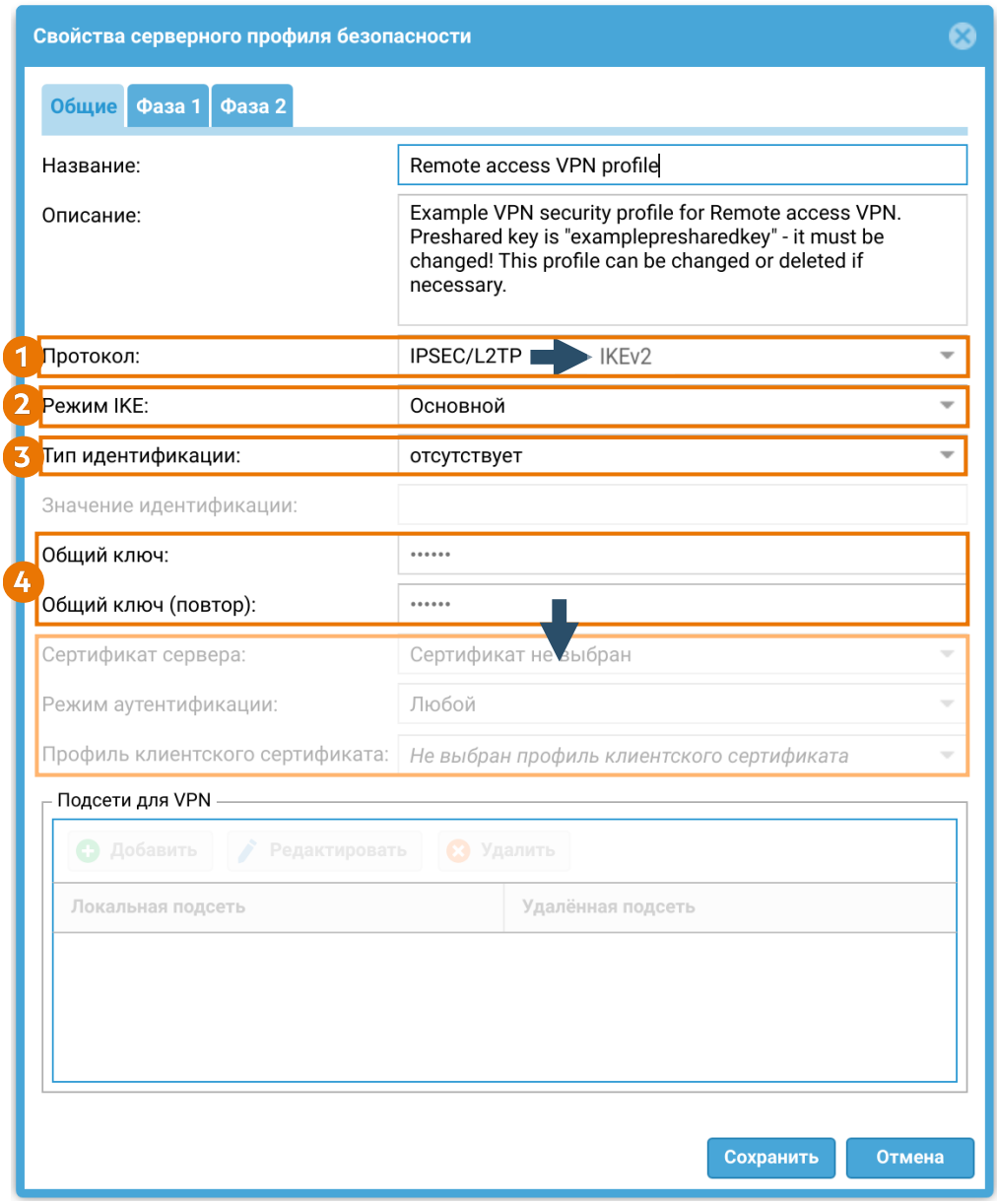
Creating a VPN Security Profile

In the VPN security profile settings, the types and settings of encryption and authentication algorithms are defined. Multiple security profiles may be used for connecting to different client types.

To create a security profile in the **VPN server's** administrator web console, go to the **VPN → Server security profiles** section:



Click the **Add** button in the section's toolbar, and fill in the required fields in security profile properties:



In the **General** tab, you can select the VPN protocol version and set authentication parameters when establishing the secure connection.

1. **Protocol**. The options are as follows:

    - **IPsec/L2TP**
    - **IKEv2**
    - **IPsec only/IKEv1**: not used for Remote Access VPN connections (it is used for Site-to-Site VPN connections only).

2. **IKE mode** (available for IKEv1 only). For more information on IKEv1 modes, please see the VPN article. The following field selection options are available:

    - **Main**.

Aggressive.

3. **ID type** (the IKE local ID parameter). This is required for DCFW identification on a neighbor node when establishing a VPN connection to certain vendors' equipment. Enumerated selection options:

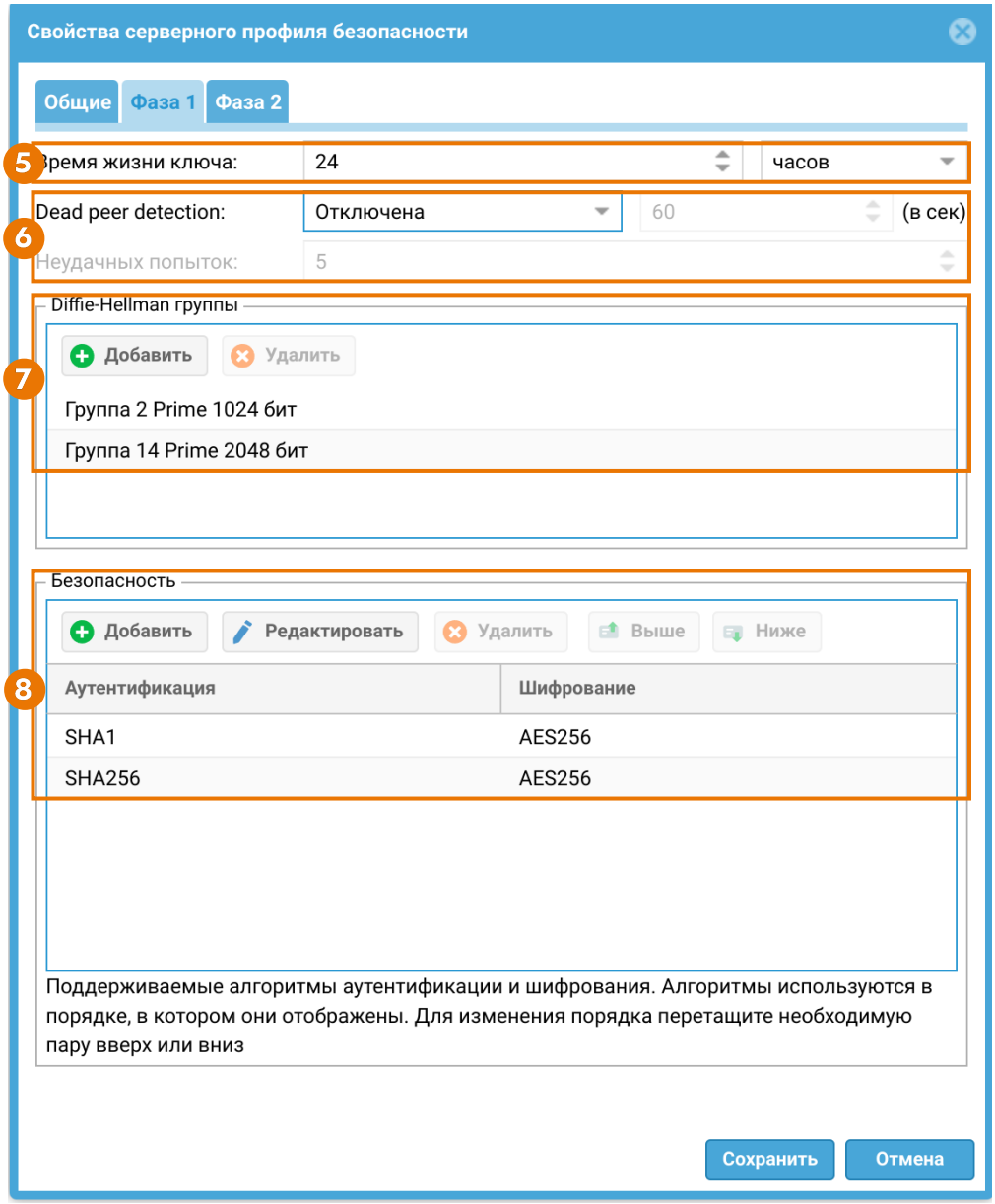
- **None**: the default value of the field. Used when the IKE local ID parameter is not required for establishing a VPN connection.
- **IPv4**: the host's IP address.
- **FQDN**: the host's address in the fully-qualified domain name (FQDN) format.
- **CIDR**: the host's address in the classless inter-domain routing (CIDR) format.
- **ID value**: the IKE local ID value in the format specified above.

4. Authentication type of the remote node when establishing the secure connection.

- If you choose an IPsec or L2TP protocol, you need to use authentication based on the **pre-shared key**. You need to specify the pre-shared key. This string must match on the VPN client and VPN server for a successful connection.
- When selecting the IKEv2 protocol to establish a tunnel, the following is specified:
  - Pre-created **server certificate**;
  - **Auth mode**: whether to authenticate using **PKI**-based certificates or the EAP protocol with the MSCHAPv2 (**AAA**) method.
    - If you select the **PKI** mode, you will need to specify a previously configured **client certificate profile** (for more information on client certificate profiles, please see the Client certificate profiles section).
    - In EAP authentication mode with the MSCHAPv2 (**AAA**) method, the client exchanges EAP packets with a VPN server. In this case the VPN server forwards the packets to an external domain RADIUS server, which then makes the authorization decision. Once the authorization is received from the RADIUS server, the VPN server requests user information from the domain server using the received login, and then makes the decision to connect the user to the VPN.

Next, the cryptographic parameters for the first and second phases of secure connection negotiation need to be configured.

In the first phase, an IKE SA is negotiated and established. Provide the following settings:



5. **Key lifetime**: the time period after which the parties re-authenticate and re-negotiate the first-phase settings.

6. **Dead peer detection** (DPD): to check that the channel is working and to disconnect/reconnect the channel if the connection is lost. DPD sends R-U-THERE messages periodically to check if the IPsec neighbor is available. There are 3 operating modes of the mechanism:

- **off**: the mechanism is disabled. DPD requests are not sent.
- **always on**: DPD requests are always sent within the specified time interval. If no response is received, additional requests are sent sequentially at intervals of 5

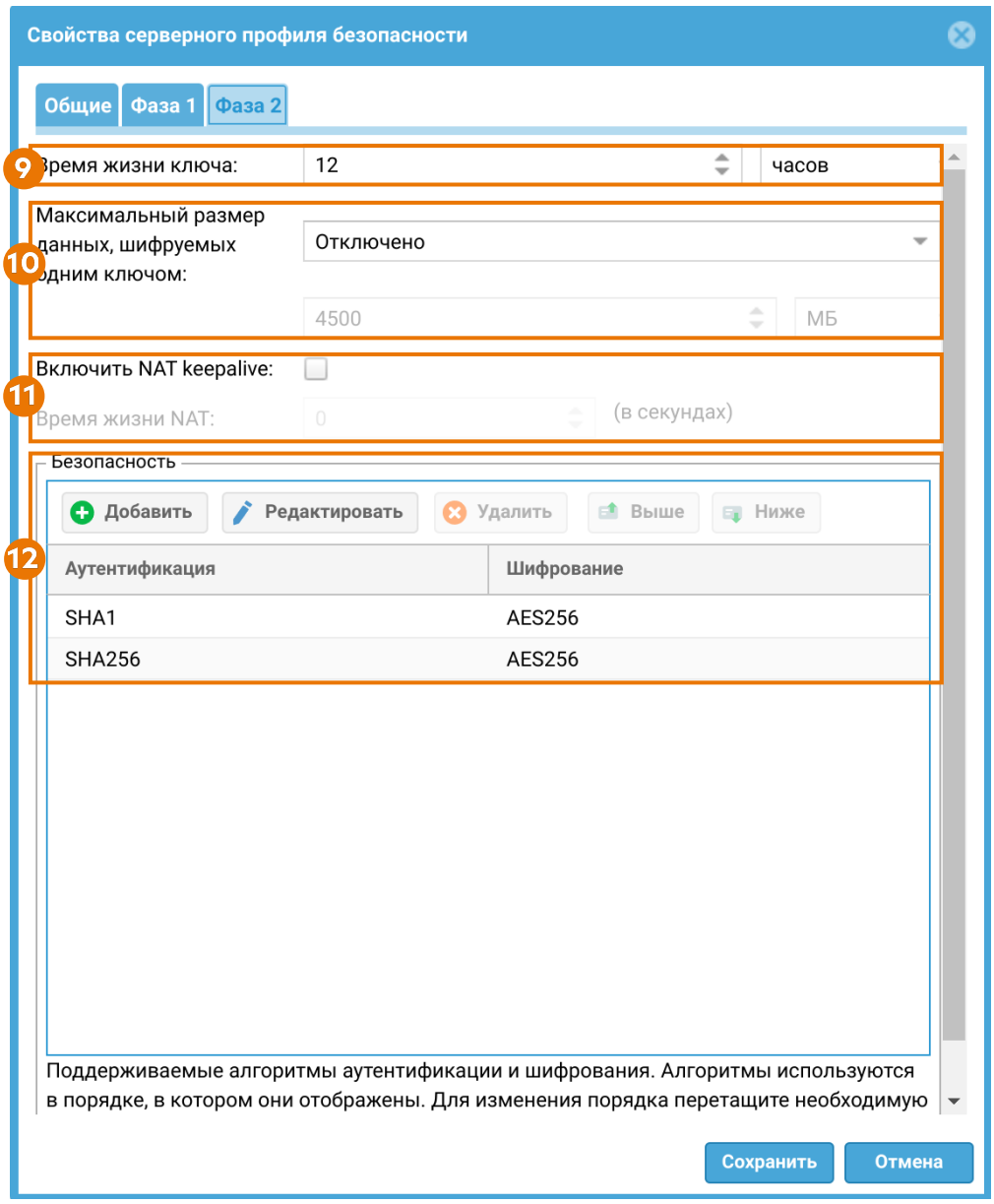
seconds in the number specified in the **Failures** field. If there is a response, the mechanism returns to the initial interval for sending DPD requests, and if there is no response, the connection is terminated.

- **Idle**: DPD requests are not sent while there is ESP traffic through the created SAs. If there are no packets within twice the specified time interval, then a DPD request is sent. If there is a response, a new DPD request will be sent again after a double interval of the specified time. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **Failures** field. If there is no response, the connection is terminated.

7. **Diffie-Hellman groups**: select the Diffie-Hellman groups that will be used for key exchange.

8. **Security**: select authentication and encryption algorithms. Algorithm are used in the order they are listed here. To reorder the algorithms, drag and drop them with the mouse or use the **Up**/**Down** buttons.

In the second phase, the method for securing data in the IPsec connections is selected. Provide the following settings:



9. **Key lifetime**: the time period after which the nodes must rotate the encryption key. The lifetime for the second phase is shorter than for the first one, which entails a more frequent key rotation.

10. **Key lifesize**: the key lifetime can also be expressed in bytes and is called lifesize in that case. If both values (**Key lifetime** and **Key lifesize**) are specified, the counter that reaches the limit first will trigger session key re-generation.

11. **NAT keepalive**: used in scenarios when IPsec traffic goes through a NAT node. NAT table entries are active for a limited time. If there was no VPN traffic over the tunnel during that time span, NAT table entries on the NAT host will be deleted, preventing further passage of VPN traffic. The VPN server located behind the NAT gateway uses NAT keepalive function to periodically send keepalive packets to a peer node in order to keep the NAT session active.

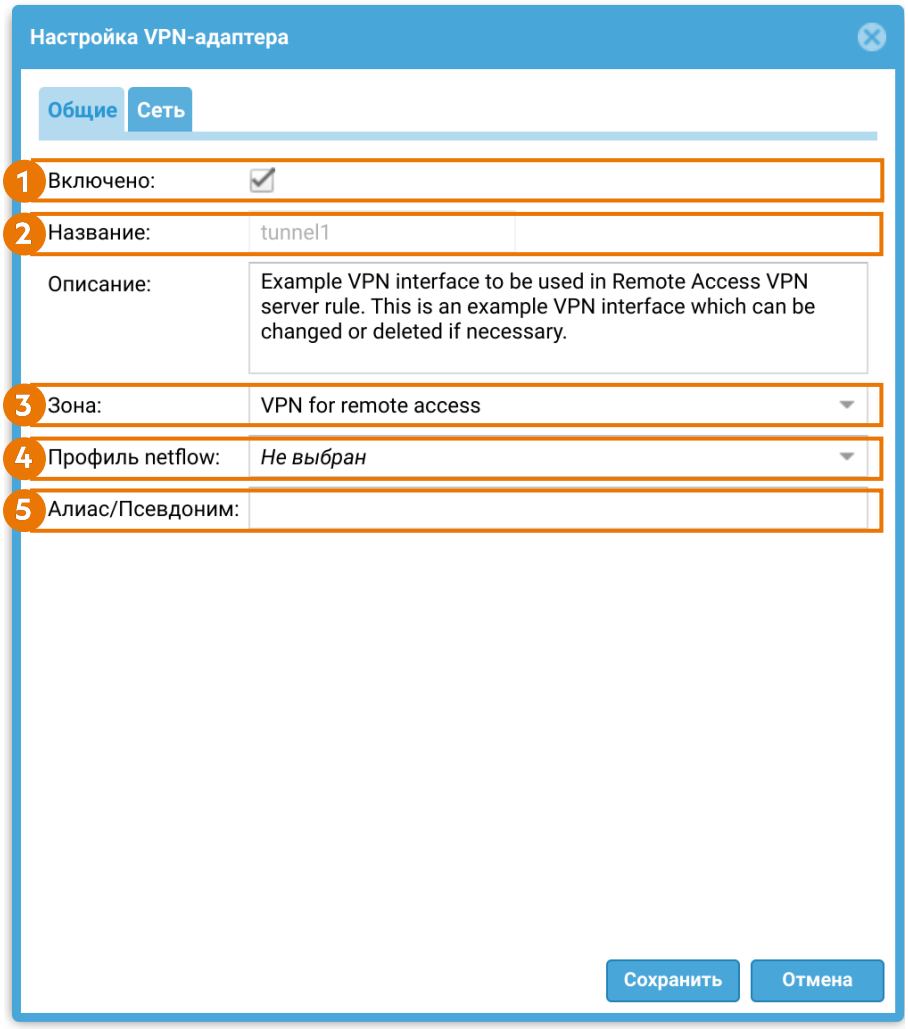
12. **Security**: the algorithms are used in their listing order. To reorder the algorithms, drag and drop them with the mouse or use the **Up**/**Down** buttons.

As an example, the **Remote Access VPN profile** with all the required settings is created in the admin web console. If you plan to use this profile, make sure to change the pre-shared encryption key when the IKEv1/IPsec protocols are used.

Creating a VPN Interface

A VPN interface is a virtual network adapter that will be used to connect VPN clients. This is a cluster-type interface, which means that it will be created automatically on all UserGate nodes included in a configuration cluster. If an HA cluster exists, in case any problems are identified with the active server, VPN clients will be automatically switched to a backup server, and without terminating existing VPN connections.

To create a VPN interface, use the **Network → Interfaces** section of the Admin console. Click the **Add** button, select **Add VPN** and specify the necessary parameters in the VPN device settings:



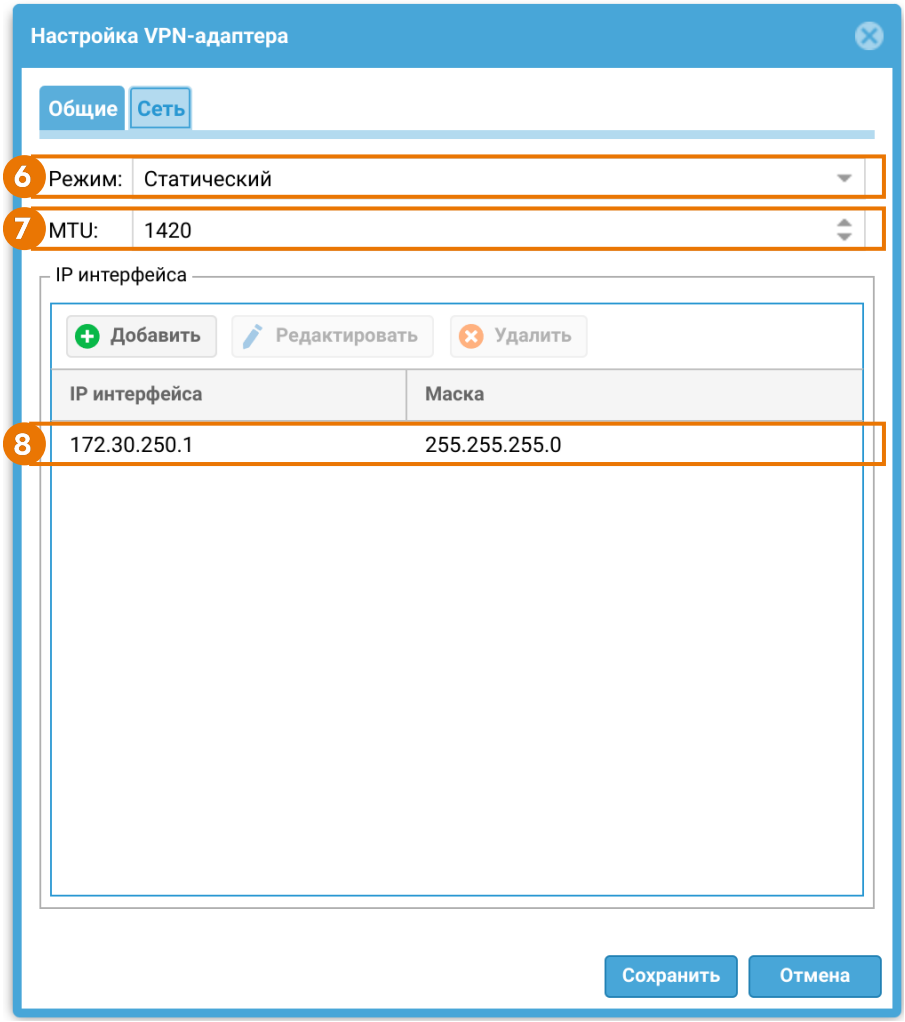
1. **Enabled**: enables or disables the interface.

2. **Name**: the name of the interface. Should be in the form *tunnelN*, where *N* is the ordinal number of the VPN interface.

3. **Zone**: the zone to which this interface will belong. All clients with a VPN connection to DCFW will be placed in this zone as well. In this field, you need to specify the zone created earlier at the stage of creating the zone for VPN connections.

4. **Netflow profile**: the Netflow profile used for this interface. For more details on Netflow profiles, see the Netflow Profiles section. (Optional)

5. **The interface's alias**. (Optional)



6. **Mode**: IP address assignment type. The options are no address, a static IP address, or a dynamic IP address obtained using DHCP. If the interface is to be used for receiving VPN connections (Remote access VPN), a static IP address must be used.

7. **MTU**: the MTU size for the selected interface. If packets transmitted over the VPN tunnel exceed the maximum MTU at any of the intermediate devices, they can be split into fragments. This can increase the latency and reduce the performance. By

setting an optimum MTU value on the tunnel interface, you can avoid packet fragmentation and reduce the latency.

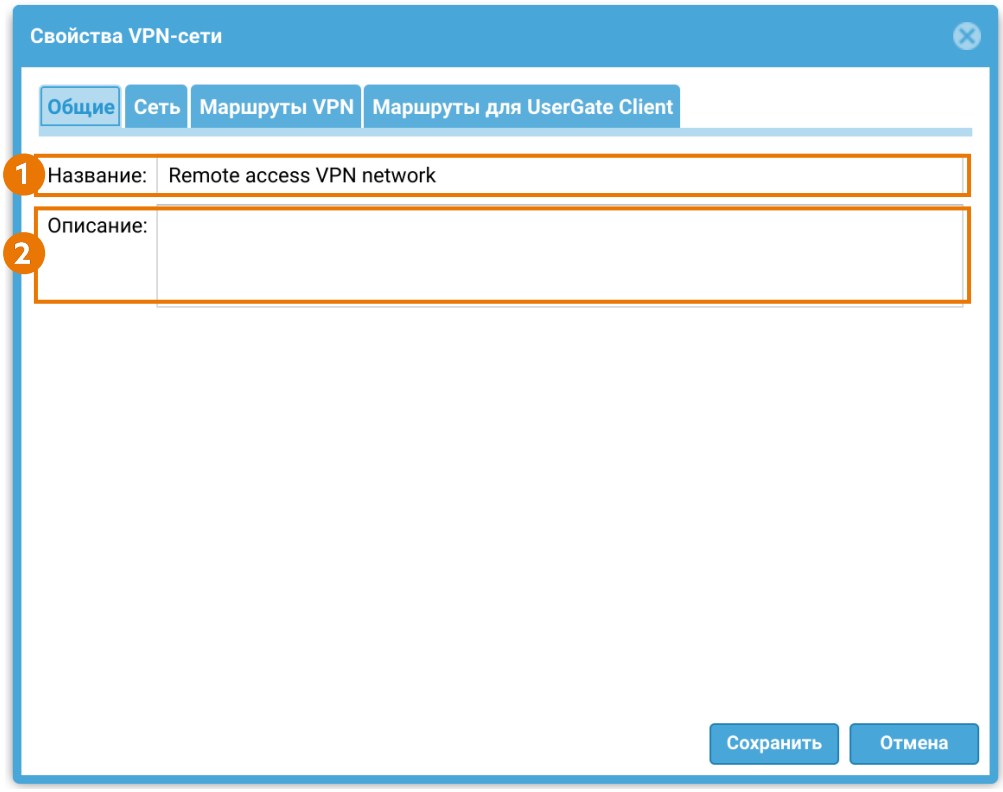
8. This field is used to specify the **IP address** of the VPN interface if a static IP address is used.

As an example, there is a predefined VPN interface named **tunnel1** in the admin web console that is recommended for use as a Remote Access VPN interface.

Creating a VPN Network

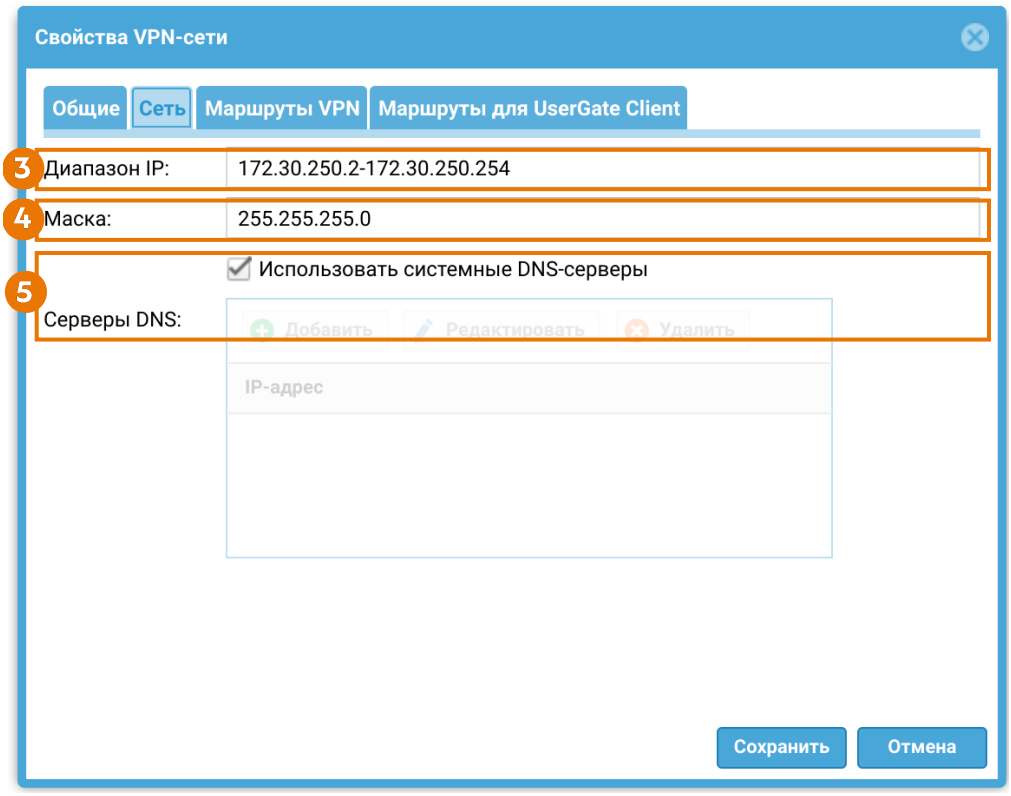
A VPN network determines the network settings that will be used for connecting the client to the server. This is primarily the assignment of IP addresses to the clients inside the tunnel, the DNS settings, and the routes that will be passed to the clients that support the use of routes assigned to them. Multiple tunnels may be used with different settings for different clients.

To create the VPN network, use the **VPN → VPN networks** section of the Admin console. You need to click **Add** and fill the necessary parameters in the VPN network properties:



1. **Name** of the VPN network.

2. A VPN network **description**. (Optional)



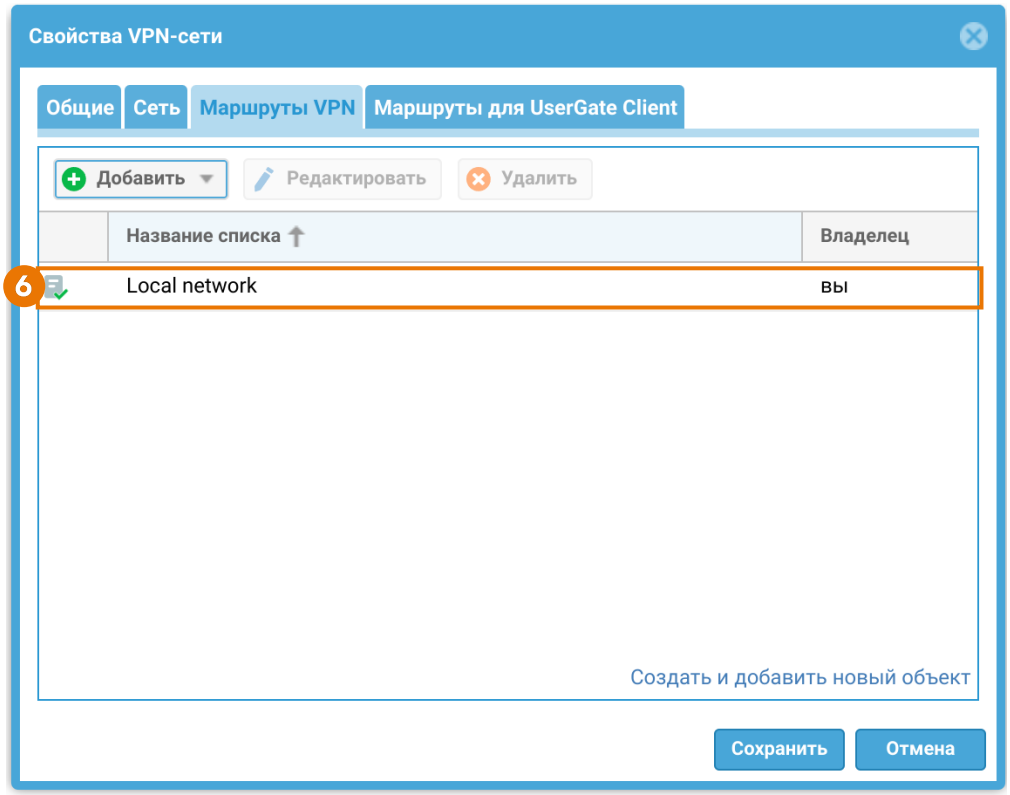
3. **IP addresses range**, which will be used by the clients. You must exclude the address assigned to the **VPN interface** of DCFW used along with this network from this range. Do not enter network addresses or the broadcast address here.

4. **Mask** of the VPN network.

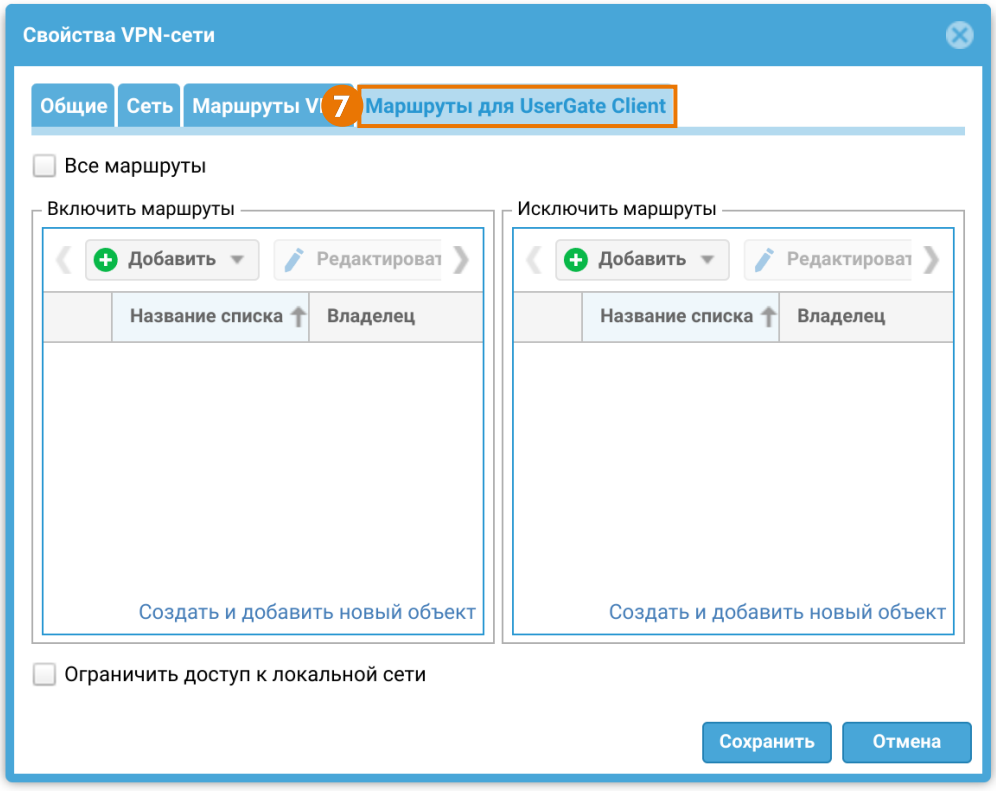
5. Specify the **DNS servers** that will be passed to the client or set the **Use system DNS** checkbox, in which case the client will be assigned the DNS servers used by DCFW.

Important!

A maximum of two DNS servers can be specified.



6. **VPN routes**: the routes sent to the VPN client in the CIDR format or a predefined IP address list.

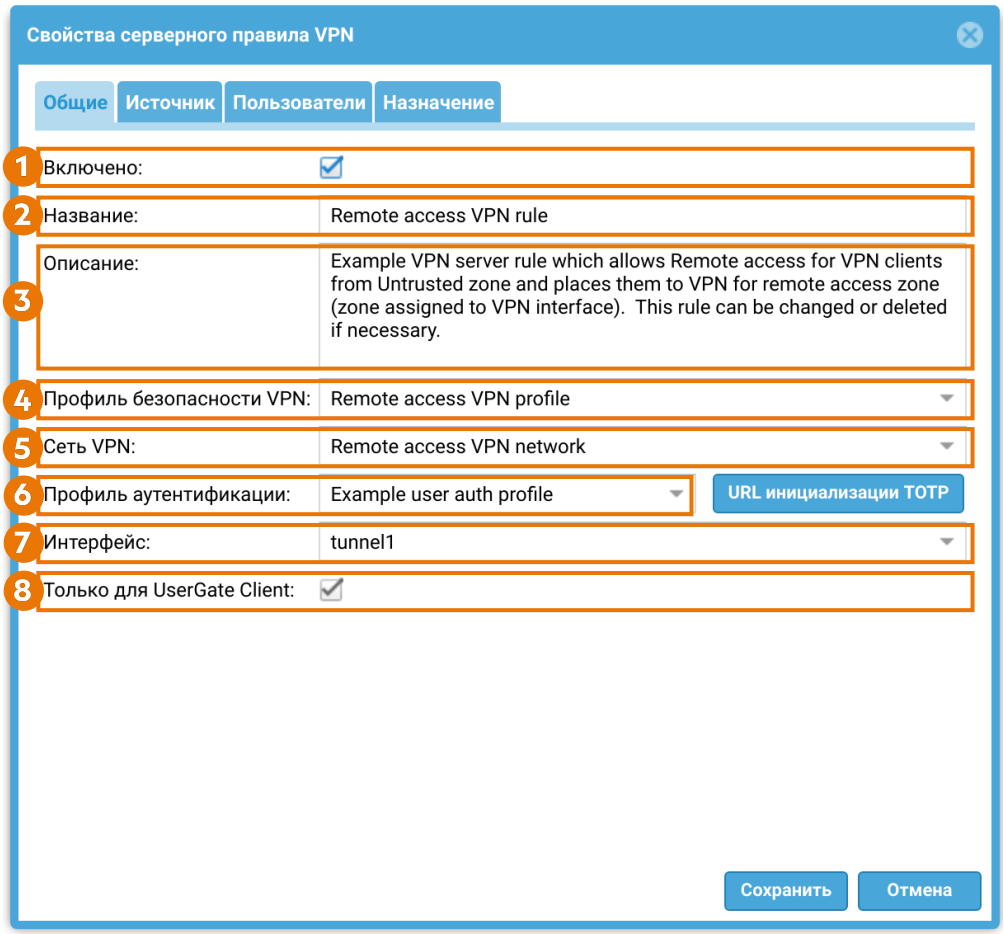


7. In the **UserGate client routes** tab, you can configure the split tunneling functionality for the UserGate Client.

As an example, a network named **Remote Access VPN network** is created in the admin web console with the default settings. To use this network, you need to add routes to it that will be passed to the client.

Creating a VPN Server Rule

To create VPN server rules, use the **VPN → Server rules** section of the Admin console. Then click **Add** and fill in the relevant fields in the rule properties:



1. **Enabled**: enables or disables the VPN rule.

2. VPN server rule **name**.

3. VPN server rule **description**. (Optional)

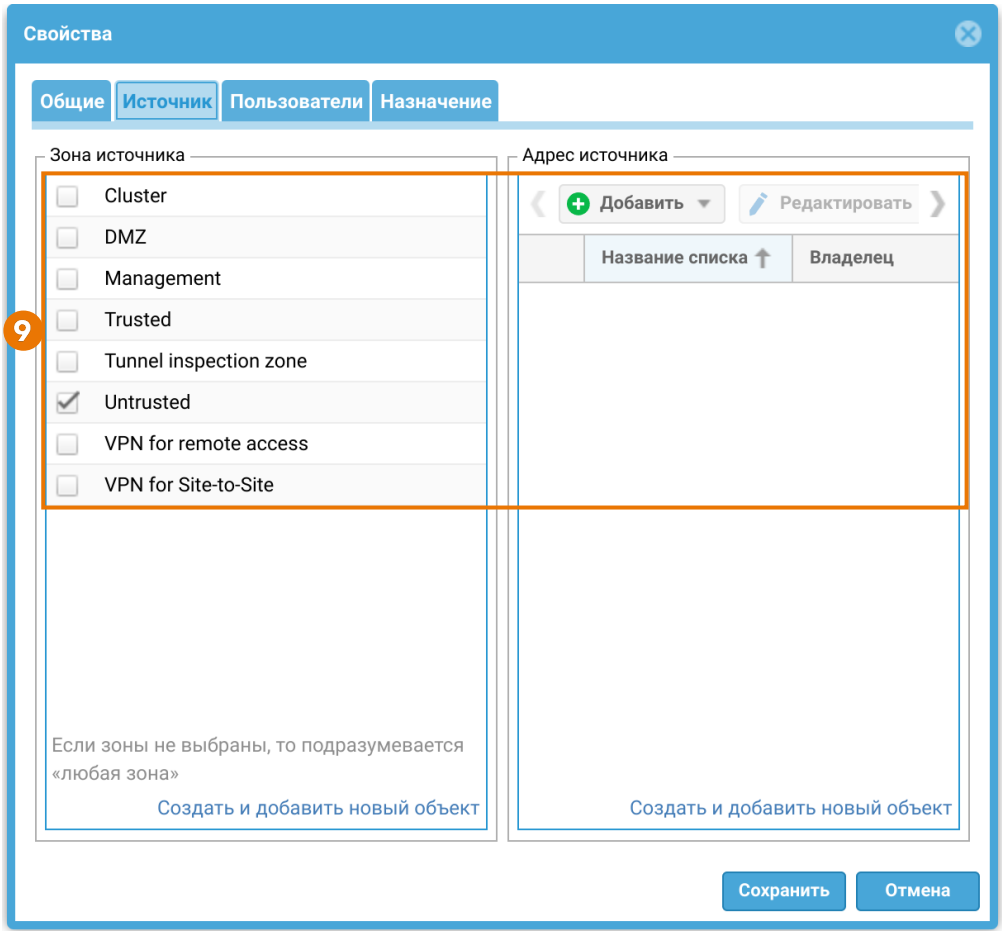
4. **VPN security profile**: the security profile that was created earlier.

5. **VPN network**: the network created earlier, at the stage of creating VPN network.

6. **Authentication profile**: the authentication profile for VPN users created earlier.

7. **Interface**: the VPN interface created earlier.

8. **For UserGate Client only**: this option (available starting from software version 7.1.2) allows you to limit the connection capability under this rule for UserGate Client VPN clients only.

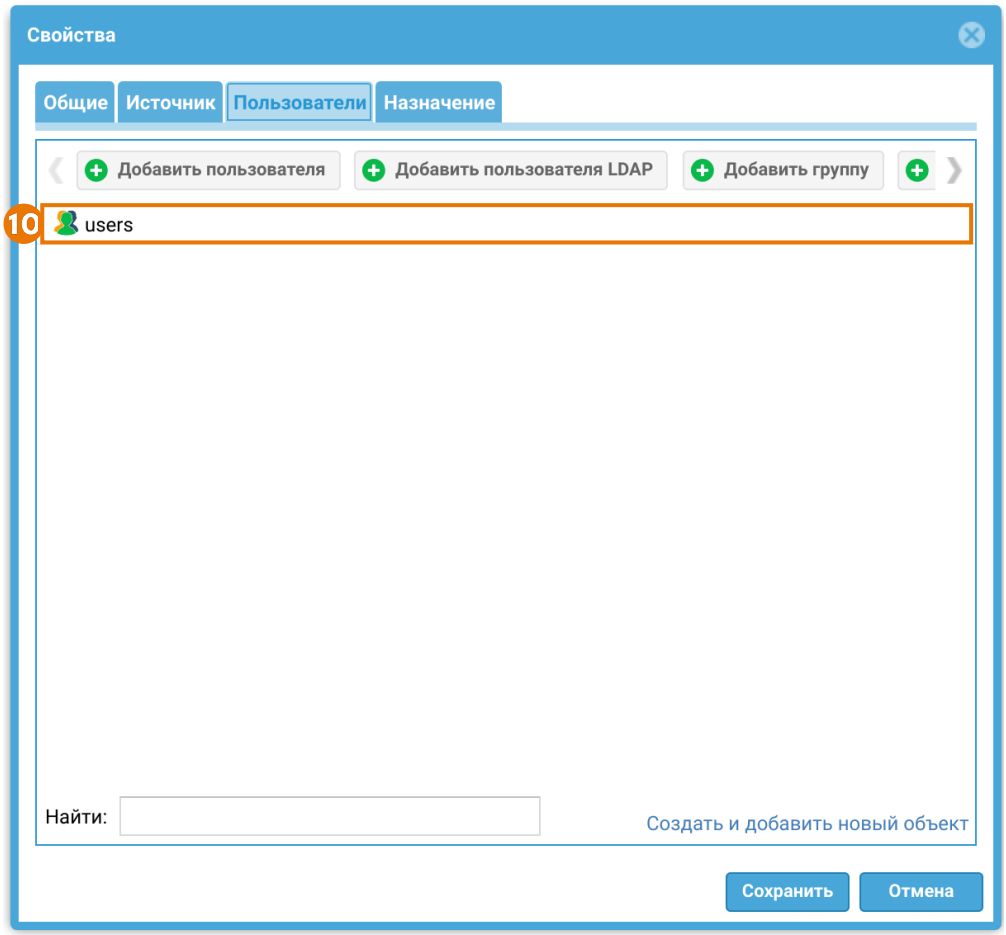


9. **Source**: the zones and IP addresses from which VPN connections are allowed. Normally, the clients are on the Internet, so specify the **Untrusted** zone.

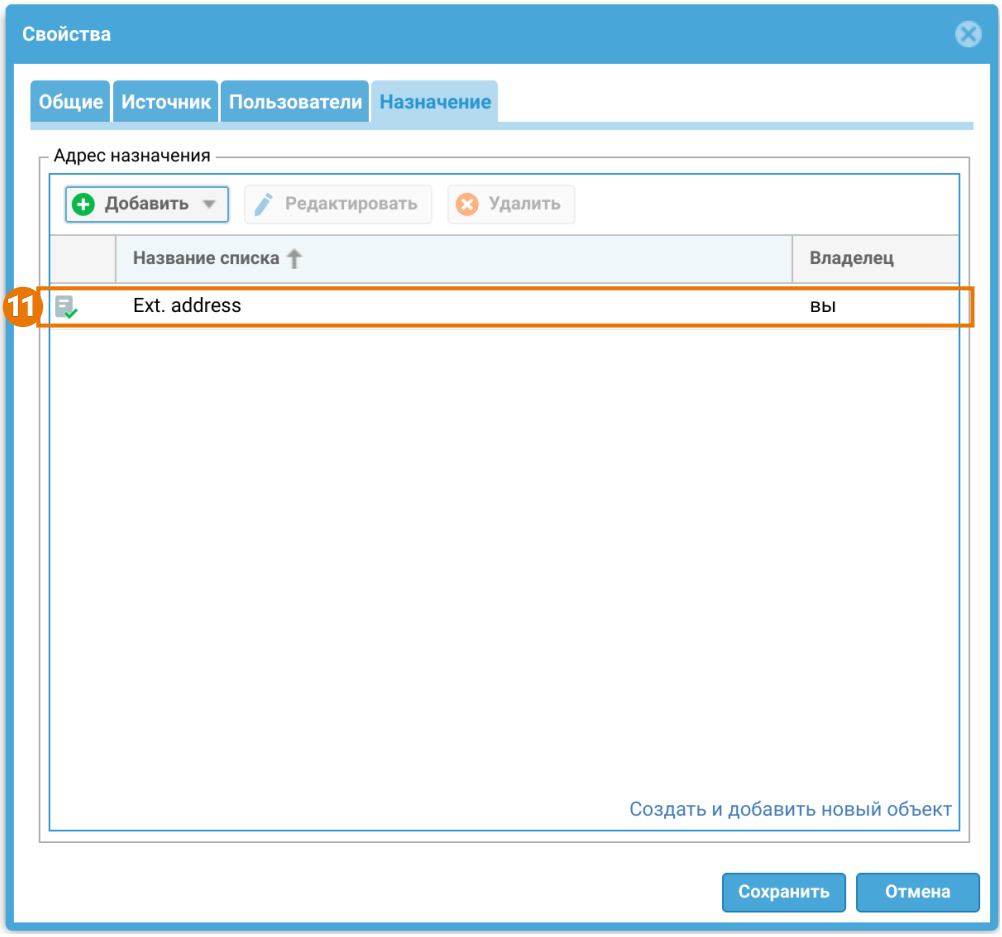
Important!

The traffic processing logic is as follows:

- **The conditions are combined using Boolean OR, if several IP address and/or domain lists are specified.**
- **The conditions are combined using Boolean AND, if GeoIPs and IP address and/or domain lists are specified.**



10. **Users**: local or domain groups of user accounts or individual user accounts that are allowed to connect via the VPN.



11. **Destination**: one or more interface addresses to which the clients will connect. The interface must belong to the zone specified at the stage of zone access control.

Important!

To apply different server rules to different clients, use the *Source zone* and *Source address* settings. The *Users* setting does not govern the selection of a server rule, as the user is checked only after the VPN connection has been established.

Note

When changing the VPN server settings (changing server rules, changing security profiles, adding new VPN networks), the VPN server does not reboot, so previously established active VPN client sessions are not terminated. A reboot of the VPN server and reconnection of active VPN client sessions may occur if the IP address of the tunnel interface of the VPN server is changed.

As an example, a network named **Remote Access VPN network** is created in the admin web console with the default settings. To use this network, you need to add routes to it that will be passed to the client.

Control of Access to Resources

Clients connect to the VPN server using the Point-to-Point protocol. To allow traffic to flow from the previously created zone for VPN connections, you need to create a NAT rule from this zone to all the required zones. The rule is created in the **Network policies → NAT and routing** section. For more information on NAT rules, see the NAT and Routing section. As an example, there is a **NAT from VPN for remote access to Trusted and Untrusted** rule created in the admin web console, allowing IP address spoofing from the **VPN for remote access** zone to **Trusted** and **Untrusted** zones.

To grant VPN users access to certain network segments or, for example, Internet, go to **Network policies → Firewall** and create a firewall rule that allows traffic from the zone for VPN connections to the desired zones. For more details on configuring firewall rules, see the Firewall section of the guide.

As an example, a rule named **VPN for Remote Access to Trusted and Untrusted** is created in the admin console that allows all traffic from the zone **VPN for Remote Access** to the **Trusted** and **Untrusted** zones. This rule is disabled by default

Configuring VPN client

Once the VPN server is configured, you need to configure the VPN clients. UserGate Client can be used as the client software on user devices, with native clients of most popular operating systems also supported.

Configuring the client software parameters depends on the type of connection to be established.

L2TP/IPsec VPN

When a VPN is created using L2TP / IPsec (IKEv1), L2TP creates a tunnel where network-layer packets are transmitted inside PPP frames. IPsec ensures encryption, authentication, and checking the integrity of the transferred data.

Mutual authentication of nodes is done by using a **pre-shared key**. VPN users are authenticated using a login and password.

The following parameters must be specified in the client software settings:

- VPN type (L2TP/IPsec);

- VPN server name or IP address;
- Pre-shared key value;
- User authentication method (PAP);
- Settings for the first and second phases of secure connection negotiation;
- User authentication parameters (login and password).

Windows 10 and later Windows versions do not support L2TP connections to servers located behind upstream NAT routers by default. To be able to establish this kind of connection, the following tweaks need to be made in the Windows registry:

- under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent**, create a **DWORD (32 bit)** key named **AssumeUDPEncapsulationContextOnSendRule** and assign it a value of **2**;
- under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Parameters**, change the value of the **AllowL2TPWeakCrypto** key to **1**.

Important!

After making changes to the registry, they need to be applied. For example, this can be done by rebooting your computer.

To learn more, read this article by Microsoft: https://docs.microsoft.com/en-US/troubleshoot/windows-server/networking/configure-l2tp-IPsec-server-behind-nat-t-device.

IPsec (IKEv2) VPN Authentication Based on the Certificates Using the Public Key Infrastructure (PKI)

When a VPN is created using IPsec(IKEv2), a secure VPN tunnel is established only using the IPsec group of protocols together with the IKE version 2 (IKEv2) protocol. During the tunnel creation process, mutual authentication of nodes occurs with verification of the authenticity of certificates.

The certificate verification process is as follows:

- Over an encrypted channel, the VPN client sends its certificate and encrypted data signed with a private key.

- The VPN server decrypts data with the client's public key and compares it with its own control set, thereby verifying if the client has the private key.
- The VPN server checks the client's certificate against the specified certificate chain, thereby verifying that the certificate was issued by an authorized CA.
- The VPN server can check if the certificate was revoked.
- The VPN server extracts the username from the certificate and searches for the user by means of the method specified in the configured authentication profile.
- If any of the specified check points fail, the connection will not be established.
- When all checks are passed, the VPN server sends its certificate and encrypted data signed with its private key.
- The VPN client then verifies if the server's signature is correct and its name in the certificate matches the address to which the client is about to connect.

When setting up the client software, you need to perform the following operations:

- Import the previously created client certificate and root authority certificate to the workstation;
- Specify VPN type (IKEv2);
- Specify the VPN server address;
- Specify the authentication method;
- Configure virtual adapter settings.

IPsec (IKEv2) VPN with EAP authentication using the MSCHAPv2 (AAA) method

When a VPN is created using IPsec (IKEv2), a secure VPN tunnel is established using solely the IPsec group of protocols along with IKE version 2 (IKEv2).

In EAP MSCHAPv2 (AAA) authentication mode, there is the IKE_AUTH exchange phase when the VPN server responds to the client informing it that authorization via EAP is required. The client exchanges several EAP packets with the VPN server. The VPN server relays the packets to the external domain RADIUS server, which makes the authorization decision. After receiving permission from the RADIUS server, the VPN server, using the received login, requests information about the user and their membership in certain groups from the domain server, after which it makes a

decision about connecting the user to the VPN. Also, during the process of establishing a connection, the VPN client checks the authenticity of the VPN server certificate.

When setting up the client software, you need to perform the following operations:

- Import the root authority certificate to the workstation;
- Specify VPN type (IKEv2);
- Specify the VPN server address;
- Specify the authentication method;
- Specify user authentication parameters (a login and a password);
- Configure virtual adapter settings.

LIBRARIES OF ITEMS

General Information

This large section contains all records, website addresses, IP addresses, templates, and other items used while configuring the UserGate DCFW rules.

Predefined library data are supplied with the product. The administrator can add the desired items during use. Some library items are non-editable because they are supplied and maintained by UserGate developers. The item libraries supplied with UserGate have an automatic update mechanism. To have the items update automatically, you need an appropriate license. For more details on product licensing, see the Licensing chapter.

Services

The Services section contains a list of common services based on the TCP/IP protocol, such as HTTP, HTTPS, FTP, and others. These services can be used in DCFW rules. A predefined list of services is supplied with the product. The

administrator can add the desired items during use. To add a new service, follow these steps:

Name	Description
Step 1. Create a service.	Click **Add** and enter the name and a description of the service.
Step 2. Specify the protocol and port.	Click **Add**, select the desired protocol from the list, and specify the destination and (optionally) source ports. To specify a port range, you can use an em dash (—), such as 33333 — 33355.

Services Groups

Here the user can manage (create, update, and delete) service object groups, or services groups. Service groups can be used to configure DCFW policies.

To create a services group:

Name	Description
Step 1. Create a group.	In the **Services groups** pane, click **Add** and specify the name and (optionally) a description of the services group.
Step 2. Add services to the group.	In the **Items** pane, click **Add** and select services to be included in the group. To add all services, click **Add all**.

IP Addresses

The "IP addresses" section contains the list of IP addresses range that can be used when creating DCFW rules. A predefined address list is supplied with the product. The administrator can add the desired items during use. To add a new address list, follow these steps:

Name	Description
Step 1. Create a list.	In the **Groups** pane, click **Add** and give a name to the IP address list.
Step 2. (Optional) Specify the list update address.	Specify the address of the server where the updatable list is stored. For more details on updatable lists, see later in this chapter.

Name	Description
Step 3. Add IP addresses.	In the **Selected group addresses** pane, click **Add** and enter the addresses. An IP address entry can be in the form of an individual IP address, IP address/subnet mask, or IP address range (192.168.1.5, 192.168.1.0/24, or 192.168.1.5-192.168.2.100, respectively).

The administrator can create custom IP-address lists and distribute them centrally to all UserGate firewalls. To create such a list, follow these steps:

Name	Description
Step 1. Create a file with the desired IP addresses.	Create a file named **list.txt** with the IP address list. The address list is written to a plain text file in a column without any punctuation. Example: `x.x.x.x` `y.y.y.y` `z.z.z.z`
Step 2. Create an archive containing this file.	Put the file in a ZIP archive named **list.zip**.
Step 3. Create a version file for the list.	Create a file named **version.txt** and specify the list version number inside it, such as 3. On each update of the list, the version number must be incremented.
Step 4. Upload the files to a web server.	Upload the **list.zip** and **version.txt** files to your website so that they can be downloaded.
Step 5. Create an IP address list and specify an update URL for it.	On each DCFW, create an IP addresses list. When creating the list, select **Updatable** as the list type and enter the address for downloading updates. DCFW will check for a new version on your website according to the configured updates download schedule. **Note** **The list URL format is http://x.x.x.x/ or ftp://x.x.x.x/.**

Name	Description
	The schedule can be configured in the list properties. The available options are: • Disabled: update checking will not be performed for the selected item • Daily • Weekly • Monthly • Every ... hours • Every ... minutes • Advanced. With the Advanced option, a crontab-like format is used where the date/time string consists of six space-separated fields. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6, where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

URL Lists

The URL lists page allows you to create URL lists to be used as black and white lists in content filtering rules.

UserGate provides its own updatable URL lists. To use the lists, an appropriate license is required. For more details on product licensing, see the Licensing chapter.

Name	Description
List of search engines without safesearch capability	The list of known search engines that do not offer the ability to block search queries related to adult content. It is

Name	Description
	recommended to block such search engines for parental control purposes.
Compliance with RU URL (Custom 460)	The list of URLs prohibited by the Ministry of Justice of the Russian Federation.
Compliance with KZ URL (Custom 487)	The unified registry of domain names, Internet URLs, and network addresses containing information that is prohibited for distribution in the Republic of Kazakhstan.
Educational institutions	The list of domain names of educational institutions in the Russian Federation.
Phishing sites	A list of phishing website URLs.
Compliance with RU RKN (URL)	The unified registry of Internet URLs containing information that is prohibited for distribution in the Russian Federation. This list is available on the website http://eais.rkn.gov.ru.
Compliance with RU RKN (domains)	The unified registry of domain names containing information that is prohibited for distribution in the Russian Federation. This list is available on the website http://eais.rkn.gov.ru.

The administrator can create custom lists and distribute them centrally to all UserGate firewalls. To create such a list, follow these steps:

Name	Description
Step 1. Create a file with the relevant URL list.	Generate a file named **list.txt** with the URL list in the following format: www.site1.com/url1 www.site2.com/url2 ... www.siteend.com/urlN
Step 2. Create an archive containing this file.	Put the file in a ZIP archive named **list.zip**.
Step 3. Create a version file for the list.	Create a file named **version.txt** and specify the list version number inside it, such as 3. On each update of the list, the version number must be incremented.
Step 4. Upload the files to a web server.	Upload the **list.zip** and **version.txt** files to your website so that they can be downloaded.
	On each DCFW, create an URL list. When creating the list, select **Updatable** as the list type and enter the address for

Name	Description
Step 5. Create a list and specify an update URL for it.	downloading updates. DCFW will check for a new version on your website according to the configured updates download schedule. **Note** **The list URL format is http://x.x.x.x/ or ftp://x.x.x.x/.** The schedule can be configured in the list properties. The available options are: • Disabled: update checking will not be performed for the selected item • Daily • Weekly • Monthly • Every ... hours • Every ... minutes • Advanced. With the Advanced option, a crontab-like format is used where the date/time string consists of six space-separated fields. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6, where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

Time Sets

Time sets allow you to define time intervals that can later be used in various DCFW rules. A predefined list is supplied with the product. The administrator can add the desired items during use. To add a new time set, follow these steps:

Name	Description
Step 1. Create a time set.	In the **Groups** pane, click **Add** and provide the name and a description for the new time set.
Step 2. Add time intervals to the time set.	In the **Group** items pane, click **Add** and add an interval. Give a name to the new interval and specify the time.

Bandwidth Pools

The **Bandwidth pools** library item defines data transmission speed values that can later be used in traffic shaping rules. For more details on traffic shaping rules, see the Traffic Shaping chapter.

A predefined list is supplied with the product. The administrator can add the desired items during use. To add a new bandwidth pool, follow these steps:

Name	Description
Step 1. Create a bandwidth pool.	Click **Add** and provide a name and description for the new bandwidth pool.
Step 2. Specify the speed.	Specify the speed in kB/sec.
Step 3. Specify the DSCP value for QoS.	This parameter is optional. If set, it will be written into each IP packet. The value range is 0 to 63.

Response Pages

Using page templates, the administrator can configure the appearance of the block page and the captive portal auth page. Different templates can be used for different filtering and captive portal rules.

DCFW comes with various types of templates. These are block page, captive portal, web portal, TOTP initialization, and other templates. They can be used as samples to create custom templates, for example, in a company's brand style or in the required language.

Name	Description
Blockpage (EN) and **Blockpage (RU)** templates	The standard block page templates in English and Russian.
Captive portal user auth (EN) and **Captive portal user auth (RU)** templates	English and Russian-language templates for user authorization on the captive portal. The templates display an authentication form (name and password). If the authentication is successful, the user is granted access to the Internet.
Captive portal user auth + policy (EN) and **Captive portal user auth + policy (RU)** templates	English and Russian-language templates for user authorization on the captive portal. The templates display an authentication form (name and password) and the terms of use for the network (user agreement) as well as requiring the user to agree to the access policy. If the authentication is successful, the user is granted access to the Internet.
Captive portal: email auth (EN) and **Captive portal: email auth (RU)** templates	English and Russian-language templates for user authorization on the captive portal that allow the user to self-register in the system with email verification. For these templates to work correctly, configure the **Notifications** section in the captive profile.
Captive portal: SMS auth (EN) and **Captive portal: SMS auth (RU)** templates	English and Russian-language templates for user authorization on the captive portal that allow the user to self-register in the system with SMS verification. For these templates to work correctly, configure the **Notifications** section in the captive profile.
Captive portal policy (EN) and **Captive portal policy (RU)** templates	English and Russian-language templates for user authorization on the captive portal. The templates do not require the name and password entry; they just display the network terms of use (user agreement) and require the user to agree to the access policy. If the user agrees, they are granted access to the Internet. For these templates to work, set **Policy accept** as the authentication method in the Captive profile.
Captive portal user session (EN) and **Captive portal user session (RU)** templates	English and Russian-language templates that allow the user to end their authorized session by visiting http://logout.captive or http://USERGATE_IP/cps.
FTP client (EN) and **FTP client (RU)** templates	English and Russian-language templates for displaying FTP server content over HTTP.

Name	Description
SSL VPN RDP (EN) and **(RU)** templates	English and Russian-language templates for displaying an authentication page when connecting to RDP resources via the web portal.
SSL VPN SSH (EN) and **(RU)** templates	English and Russian-language templates for displaying an authentication page when connecting to SSH resources via the web portal.
TOTP INIT PAGE (EN) and **T OTP INIT PAGE (RU)** templates	English and Russian-language templates for displaying the TOTP device initialization page for VPN users.

To create a custom template, follow these steps:

Name	Description
Step 1. Export an existing default template.	Select one of the existing templates, click **Export**, and save the template to a file.
Step 2. Customize the exported template.	Use an editor to customize the template. Specialized HTML editors are not recommended for this purpose because they can corrupt the internal structure of the template. Use simple text editors.
Step 3. Create a new template.	Click **Add**, select the corresponding template type, give a name to the new template, and save it.
Step 4. Import the template customized at Step 2.	Highlight the newly created template, click **Import**, and select the customized template file.

Applications

Application signatures are a collection of semantic expressions describing characteristics of specific network applications. They are used in the firewall to analyze traffic at OSI Layer 7 to control network traffic.

Application Signature Types

UserGate can use two types of application signatures:

- Proprietary application signatures

- Customized application signatures

Proprietary application signatures are created by UserGate developers and automatically added to the system library if there is an appropriate license. In the list of signatures in the library, such signatures are marked as @UserGate in the **Owner** column.

Custom application signatures are created by the user. To create a custom application signature in the administrator web console, go to **Libraries → Applications** and click **Add**. After that, specify the signature properties and describe its characteristic features using the UASL syntax.

Fill in the following fields:

Name	Description
Type	Signature type: • Application • Protocol • Support — supplementary signature.
Id	Signature ID. If the field is left empty, a free ID from the user pool will be issued.
Name	The name of the signature.
Description	Signature description.
Threat level	Threat level defined by the signature. The following values are defined: • **1**: very low • **2**: low • **3**: medium • **4**: high • **5**: very high
Technology	Application technology: • **browser-based**: browser-based web application • **client-server**: client-server application • **network-protocol**: network protocol • **peer-to-peer**: peer-to-peer application

Name	Description
Category	A signature category is a group of signatures that have common parameters. The list of categories can be extended. • Media streaming • Email • Coin Miners • Tunneling • Games • Remote access • Conferencing • Trojan Horses • Business • Mobile • Proxies and anonymizers • Standard networks • VOIP • Web posting • Software update • File storage and backup • Web browsing • File sharing P2P • Instant messaging • Social networking
UASL	Description of the signature features using the UASL syntax.

Application signatures depending on the protocol type

Some application signatures require certain protocol signatures to work in the application profiles.

Such dependencies are provided in the table below:

Protocol signature	IDs of dependent signatures
SSL/TLS (id=19)	185, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 198, 199, 200, 201, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258,

Protocol signature	IDs of dependent signatures
	259, 260, 261, 262, 263, 264, 265, 267, 268, 269, 270, 271, 272, 273, 275, 276, 277, 278, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 437, 439, 440, 441, 443, 444, 445, 446, 449, 450, 451, 458, 459, 465, 466, 470, 471, 472, 474, 475, 477, 481, 482, 485, 486, 487, 490, 492, 494, 495, 496, 501, 502, 504, 505, 511, 512, 513, 515, 516, 517, 518, 521, 524, 525, 526, 527, 528, 531, 532, 535, 536, 537, 538, 539, 544, 549, 550, 552, 554, 556, 557, 560, 563, 564, 566, 567, 568, 576, 577, 579, 581, 585, 589, 590, 592, 595, 596, 597, 600, 601, 603, 604, 606, 607, 610, 612, 613, 617, 621, 622, 623, 625, 627, 632, 635, 636, 638, 710, 730, 731, 734, 738, 739, 744, 746, 748, 752, 753, 754, 755, 756, 759, 760, 761, 762, 763, 766, 769, 770, 771, 772, 773, 774, 775, 776, 781, 783, 785, 788, 790, 795, 797, 800, 801, 807, 808, 810, 811, 813, 815, 817, 818, 820, 822, 825, 826, 831, 832, 833, 835, 836, 837, 841, 842, 846, 847, 848, 850, 851, 852, 853, 854, 858, 859, 860, 863, 864, 867, 869, 872, 874, 875, 877, 878, 879, 880, 883, 885, 887, 888, 891, 893, 894, 895, 897, 898, 899, 902, 903, 904, 905, 908, 909, 1967, 2027, 4062, 4082, 4437, 4459, 5294, 5301, 5317, 5321, 5323, 5324, 5385, 5395, 5407, 5431, 5506, 5637, 5641, 5644, 5645, 5649, 5650, 5652, 5654, 5656, 5658, 5668, 5671, 5673, 5674, 5675, 5676, 5678, 5679, 5680, 5681, 5688, 5692, 5693, 5695, 5699, 5710, 5711, 5715, 5719, 5730, 5736, 5739, 5740, 5742, 5744, 5750, 5762, 5765, 5769, 5770, 5773, 5776, 5777, 5778, 5786, 5791, 5792, 5794, 5795, 5800, 5804, 5809, 5810, 5812, 5813, 5815, 5816, 5820, 5822, 5823, 5825, 5826, 5827, 5828, 7688, 7689, 7690, 7691, 7692, 7694, 7695, 7698, 7699, 7704, 7705, 7707, 7708, 7740, 7843, 7864, 7865, 7867, 7868, 8000, 8001, 8002, 8003, 8004, 8005, 8006, 8007, 8009, 8010, 8011, 8012, 8014, 8015, 8016, 8017, 8018, 8019, 8022, 8024, 8026, 8027, 8028, 8031, 8032, 8033, 8034, 8035, 8036, 8037, 8038, 8039, 8040, 8041, 8043, 8044, 8045, 8048, 8049, 8053, 8054, 8055, 8056, 8057, 8058, 8059, 8060, 8063, 8064, 8066, 8067, 8069, 8070, 8071, 8075, 8077, 8078, 8079, 8080, 8081, 8082, 8083, 8084, 8086, 8087, 8088, 8089, 8090, 8091, 8092, 8094, 8095, 8096, 8098, 8099, 8101, 8102, 8103, 8104, 8105, 8106, 8107, 9003, 9007, 9008, 9016, 9019, 9030, 9042, 9044, 9048, 9050, 9051, 9052, 9053, 9054, 9055, 9056, 9057, 9058, 9059, 9060, 9061, 9062, 9063, 9064, 9065, 9066, 9067, 9068, 9069, 9071, 9072, 9074, 9075, 9076, 9077, 9078, 9079, 9080, 9081, 9083, 9084, 9085, 9086, 9087, 9088, 9089, 9090, 9091, 9092, 9094, 9096, 9097, 9098, 9099, 9100, 9101, 9102, 9103, 9104, 9105, 9114, 9128, 9141, 9147, 9148, 9150, 9529, 9543, 9544, 9553, 9563, 9566, 9572, 9573, 9575, 9579, 9580, 9622, 9625, 9627, 9628, 9641, 9650, 9655, 9657, 9714, 9733, 10514, 11011, 11024, 11025, 11044, 11504, 12001, 12002, 12003, 12006, 12007, 12008, 12009, 12010, 12011, 12012, 12013, 12014, 12015, 12016, 12017, 12018, 12019, 12020, 12021, 12022, 12023, 12024, 12025, 12026, 12027, 12028, 12033, 12034, 12035, 12036, 12044, 12045, 12501, 14002, 14003

Protocol signature	IDs of dependent signatures
HTTP (id=3)	196, 239, 261, 475, 532, 535, 610, 612, 627, 710, 1967, 4133, 4340, 4441, 5323, 5395, 5506, 5655, 5672, 5674, 5676, 5693, 5728, 5730, 5750, 5754, 5763, 5769, 5770, 5773, 5778, 5788, 5792, 5823, 5830, 7867, 8002, 8013, 8048, 9777, 9823, 9824, 9845, 11027, 12032, 12033
DNS (id=5)	1967, 5395, 5672, 5815
IKE (id=11041)	11056

Related signatures

Some signatures depend not only on the protocol signature, but also on the related signatures.

The list of the related application signatures is provided in the table below:

Signature ID	Signature name	Depends on the protocol	Related to the signature	Note
218	Yandex.Disk	SSL/TLS (id=19)	—	For this signature to work, it is needed to add only the protocol signature (id=19) to the profile.
7707	Yandex.Disk download	SSL/TLS (id=19)	Yandex.Disk (id=218), Yandex Services (id=12044)	For this signature to work, you need to add the protocol signature (id=19) to the profile, as well as the associated signatures (id=218 and id=12044).
7708	Yandex.Disk upload	SSL/TLS (id=19)	Yandex.Disk (id=218), Yandex	For this signature to work, you need to add

Signature ID	Signature name	Depends on the protocol	Related to the signature	Note
			Services (id=12044)	the protocol signature (id=19) to the profile, as well as the associated signatures (id=218 and id=12044).
12044	Yandex Services	SSL/TLS (id=19)	—	**For this signature to work, it is needed to add only the protocol sign ature (id=19) to the profile.**
16020	Yandex Tracker	SSL/TLS (id=19)	Yandex Services (id=12044)	For this signature to work, it is needed to add the protocol signature (id=19) and the related signature (id=12044) to the profile.
12045	Yandex Cloud	SSL/TLS (id=19)	Yandex Services (id=12044)	For this signature to work, it is needed to add the protocol signature (id=19) and the related signature (id=12044) to the profile.

Applications Profiles

Purpose of an Application Profile

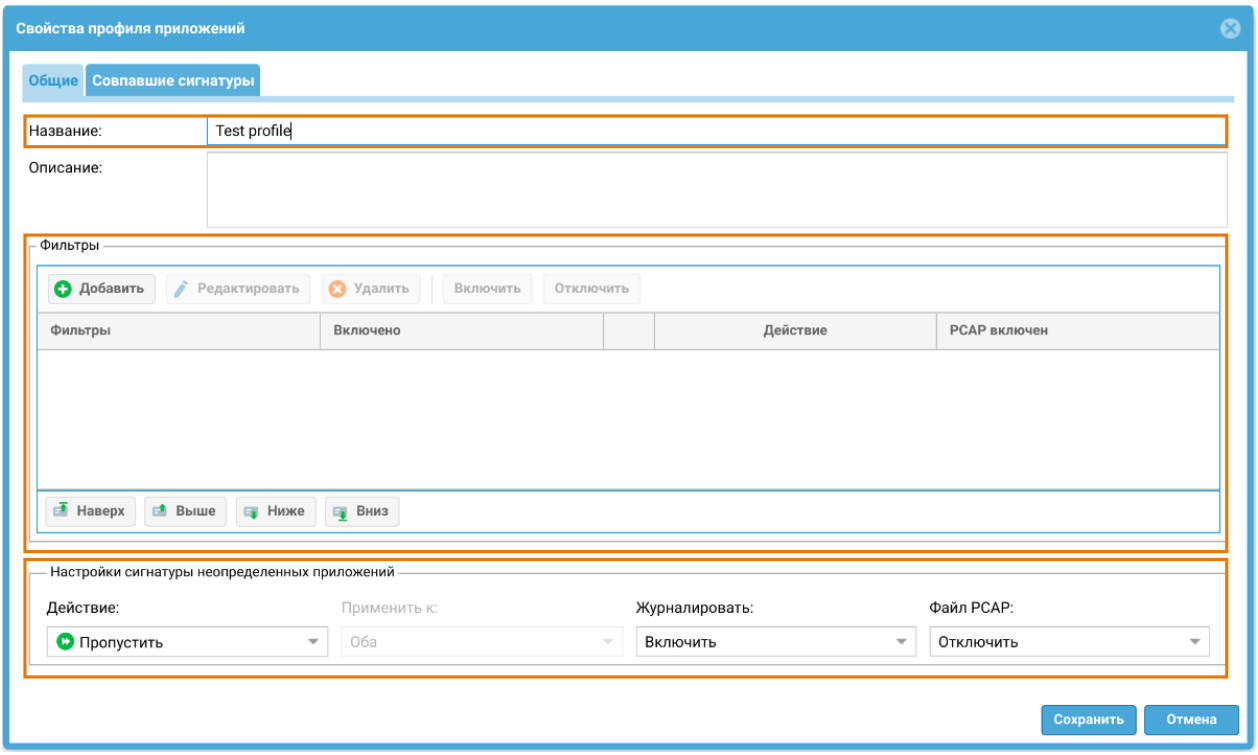
An application profile allows you to create a dynamic set of application signatures designed to analyze traffic at layer 7 of the OSI model. The dynamism of the profile is achieved due to the fact that the profile does not explicitly contain any signatures, but contains filters with which the set of signatures is collected. When the library of application signatures changes, the profiles will dynamically collect new sets of signatures that satisfy the profile filters.

In addition to creating the required set of signatures, the profile can define actions that must be performed on applications filtered by signatures and actions that must be applied to traffic that could not be identified.

Creating an Application Profile in the Administrator Web Console

In the administrator web console, application profiles are created in the **Libraries → Application profiles** section.

Click **Add** and fill in the appropriate fields in the application profile properties:



1. In the **Name** field, specify the name of the profile being created.

2. In the **Description** field, optionally specify the purpose of the profile.

3. In the **Filters** area, add filters to select the required signatures from the library and configure the actions to be performed on applications filtered by the signatures.

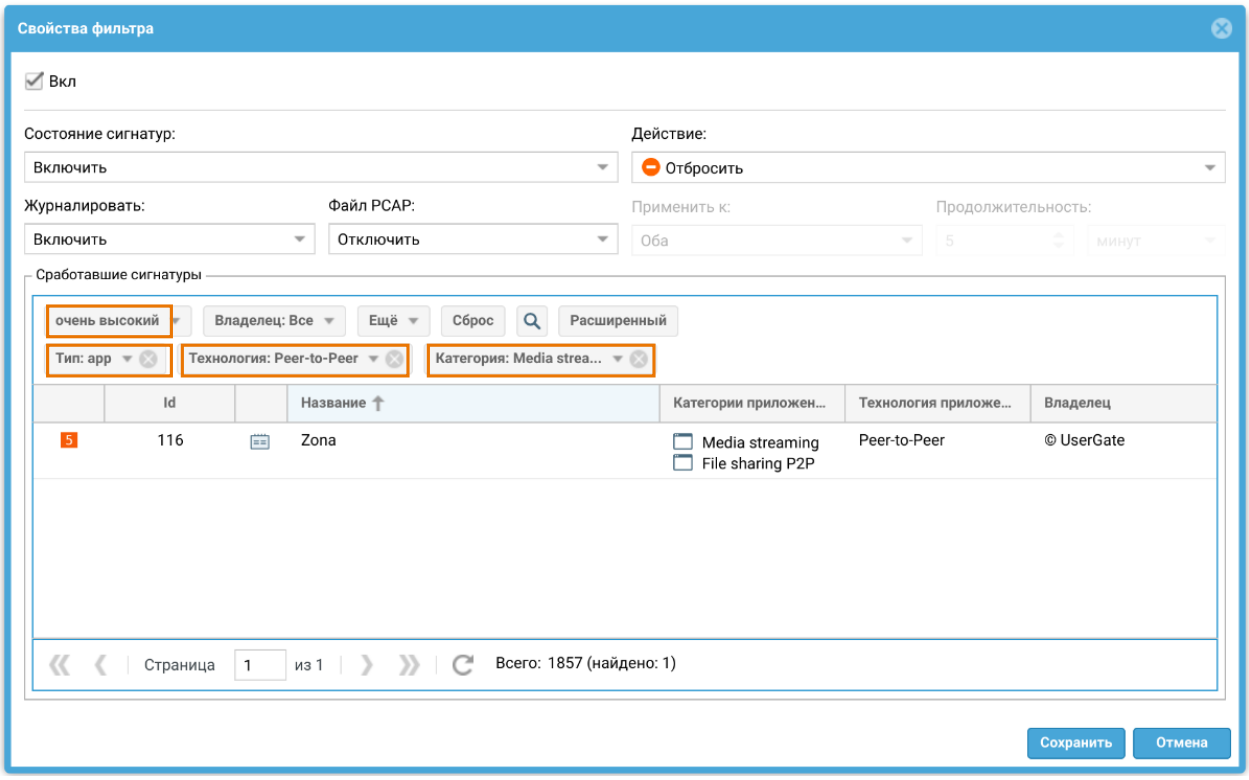
4. In the **Undefined application signature settings** area, define the actions to be performed on traffic that was not detected by the signatures of this profile.

5. The **Matched signatures** tab displays a preview of the application signatures selected by all the profile filters and the configured actions to be performed on applications filtered by these signatures.

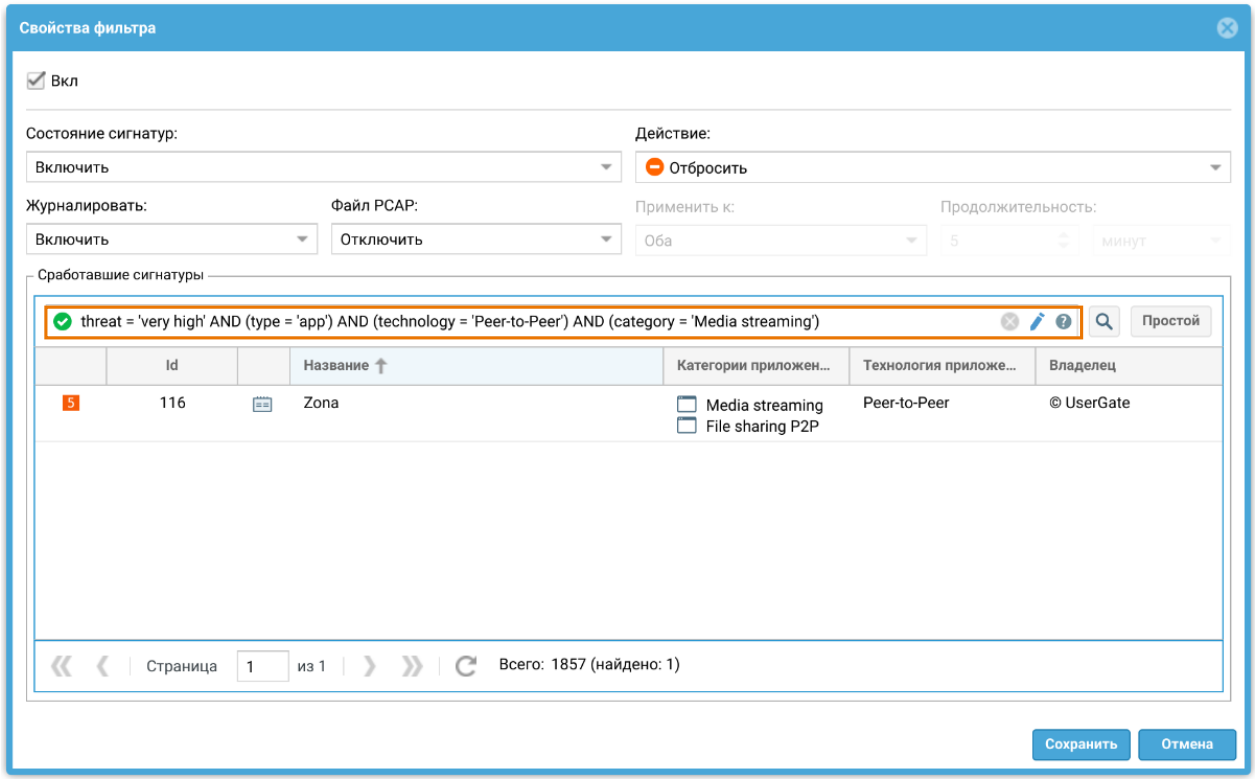
Configuring Signature Filters in the Application Profile

To create an application signature filter, click the **Add** button in the **Filters** area. The filter properties window will open.

You can create a filter by selecting the selection options in the toolbar. The window below will display the signatures from the library that fall under this filter:

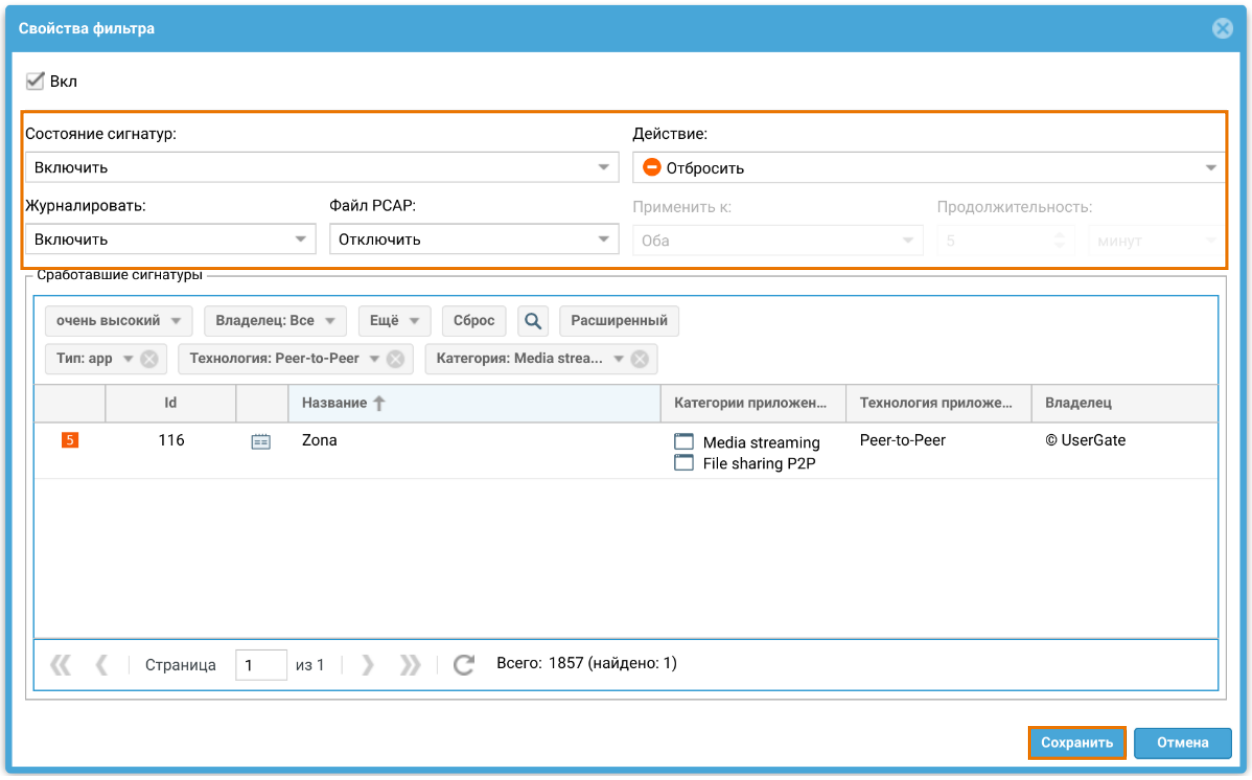


You can also create a filter by describing it using SQL-like syntax. To do this, click the **Advanced** button on the toolbar and describe the filter selection properties in the line that opens:



Each filter can configure signature states and actions that apply to all matching signatures:

- Enabling/disabling the signature
- Enabling/disabling signature logging
- Recording in pcap file if the signature is triggered
- Action taken on traffic if the signature is triggered, i.e. the application is found in the traffic. Possible actions: pass packet, drop packet, drop packet with TCP connection break, block source and/or destination IP address.



To save the created filter, click the **Save** button.

You can use several filters at once in one profile.

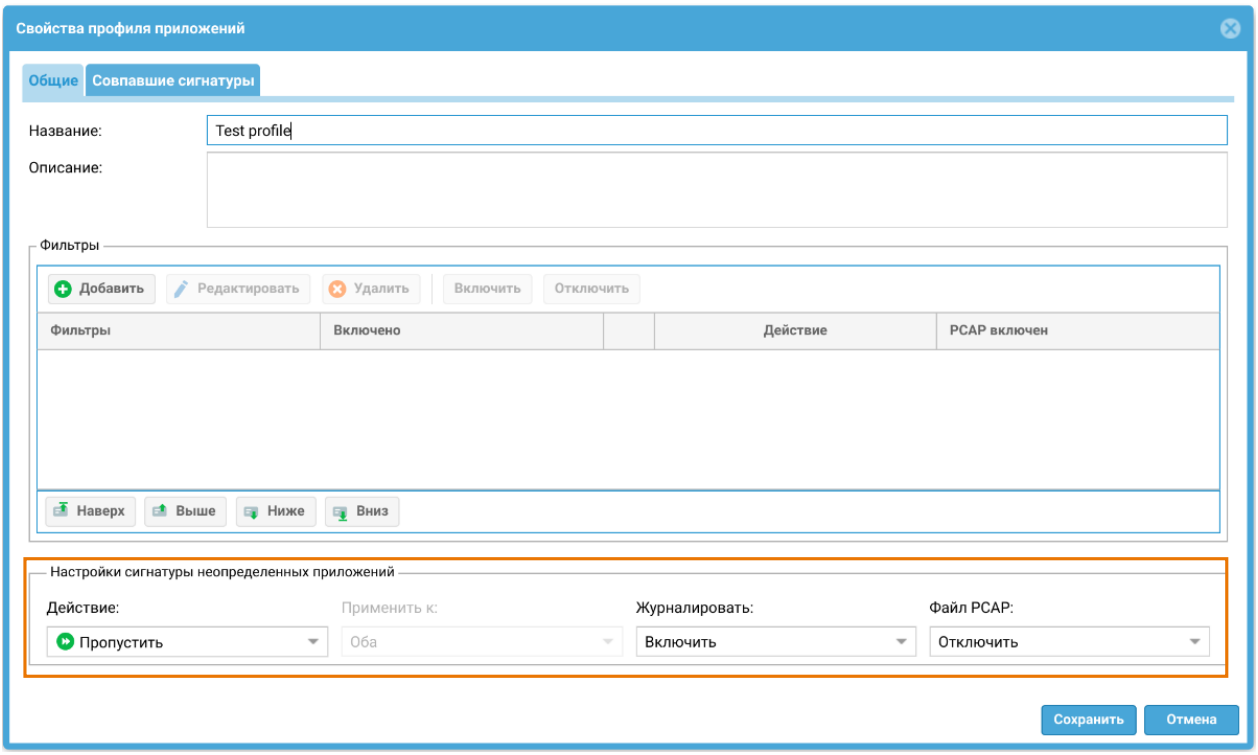
Filters in a profile use the logical OR.

The order of signature filters in a profile is important: the top filter settings have the highest priority. For example, if new signatures are added to the application signature library and they match several filters of the same profile, they will be assigned the configured action of the first filter they match.

Configuring Actions for Traffic That Could not be Identified

In an application profile, you can configure an action that is applied to traffic that could not be identified using the profile's signature set.

In the **Undefined application signature settings** area, you can configure the action, enable/disable logging and writing to the pcap file.



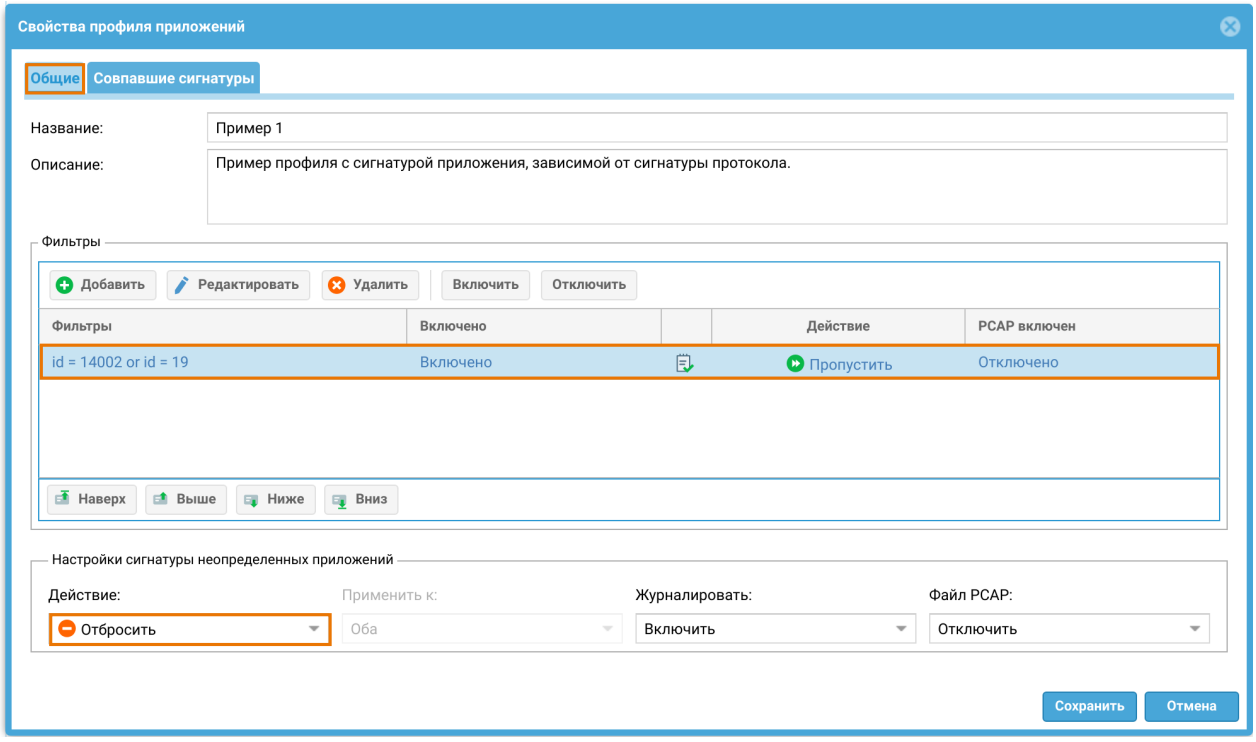
Possible actions: pass packet, drop packet, drop packet with TCP connection break, block source and/or destination IP address.

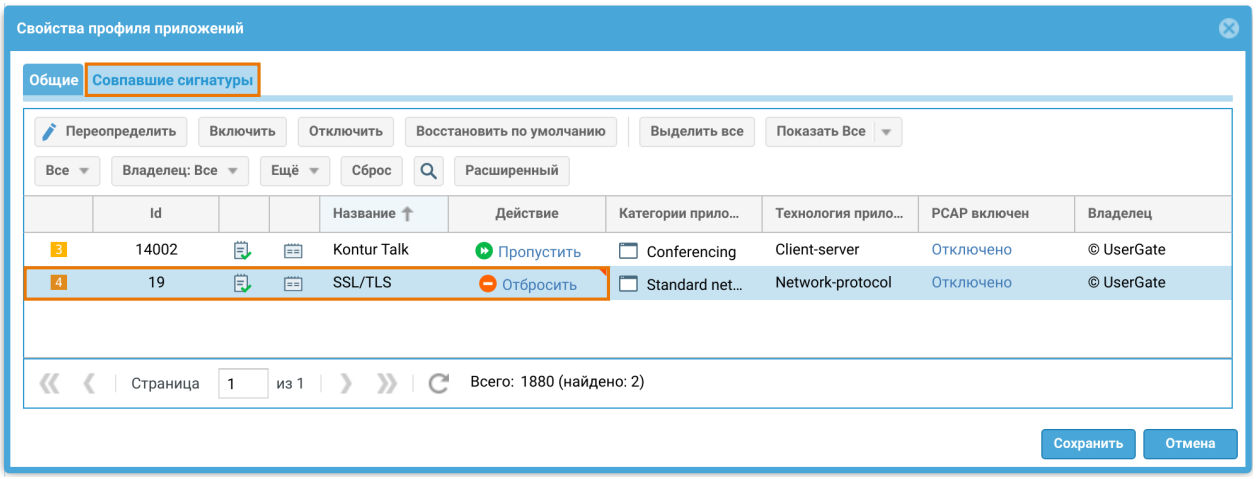
Examples of Application Profile Settings

Example 1. Application profile with a signature dependent on a protocol signature

Lists of signatures dependent on protocol signatures are provided in the Applications section.

Let's create a profile for the Kontur Talk application, which is defined by the corresponding signature (id=14002). To block all traffic except for Kontur Talk application traffic, the application profile should look like this:



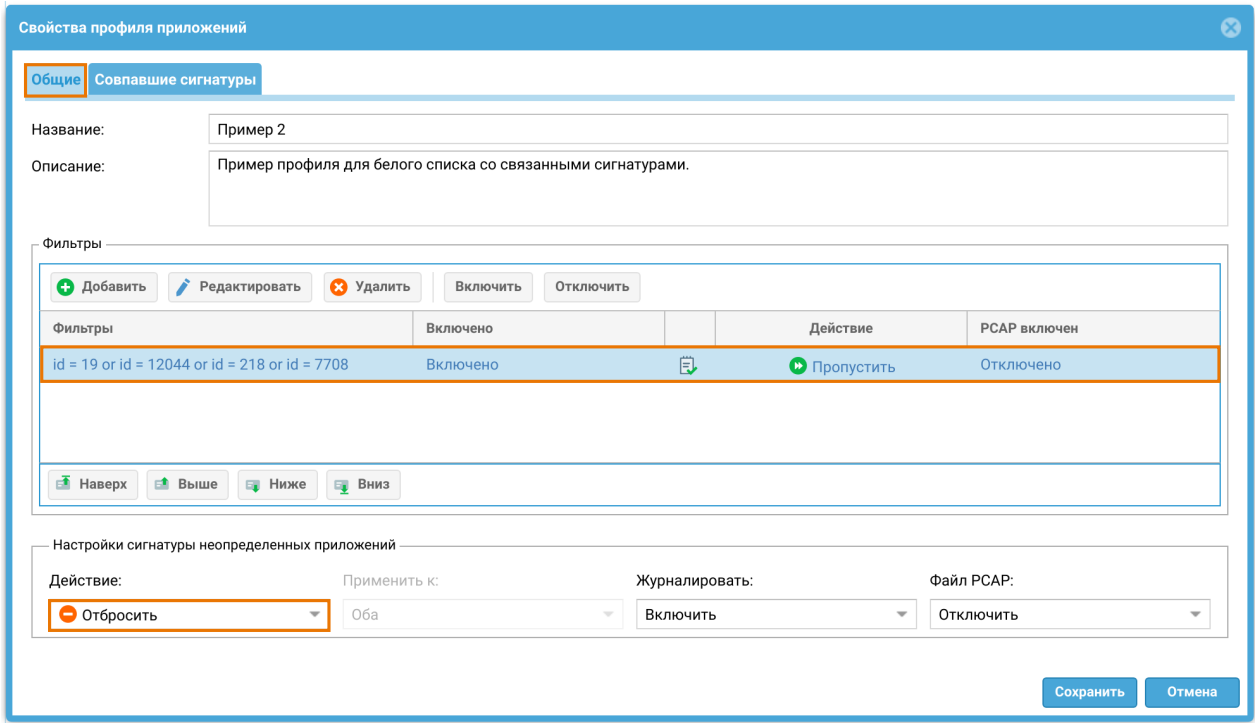


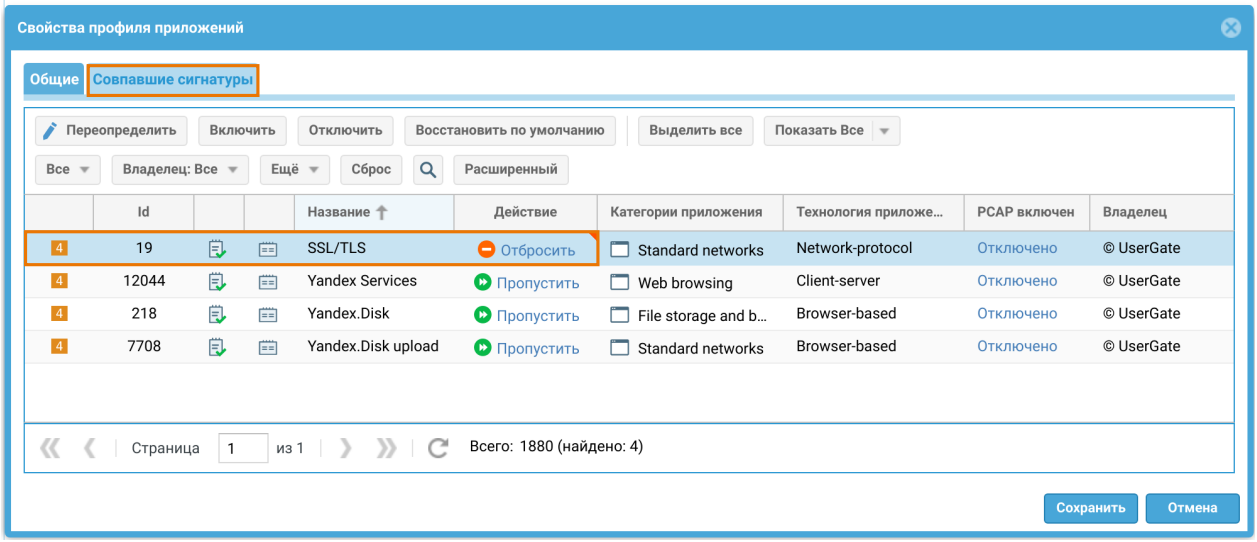
The SSL/TLS signature (id=19) is needed for the Kontur Talk signature to work. That is why it is added to the profile, but the **Drop** action is specified for it, to reject the foreign SSL/TLS traffic. The **Drop** action is also specified for the non-identified traffic.

Example 2. Whitelist profile with associated signatures

For the lists of related signatures see the Applications section.

Let's create the profile for the case when it is needed to allow uploading files to Yandex Disk. In order for the Yandex.Disk upload signature to work, we need to take into account its dependency on the SSL/TLS (id=19) protocol signature and its relation to the Yandex.Disk (id=218) and Yandex Services (id=12044) signatures. In this example the application profile will look as follows:

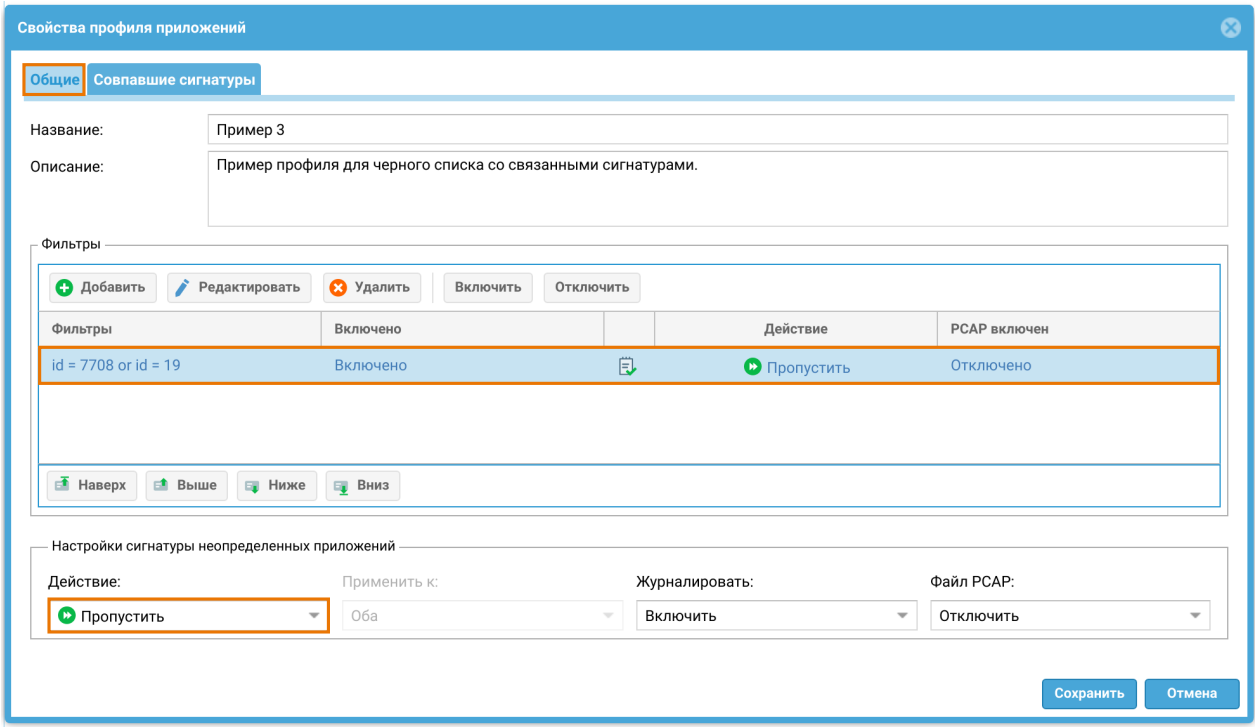


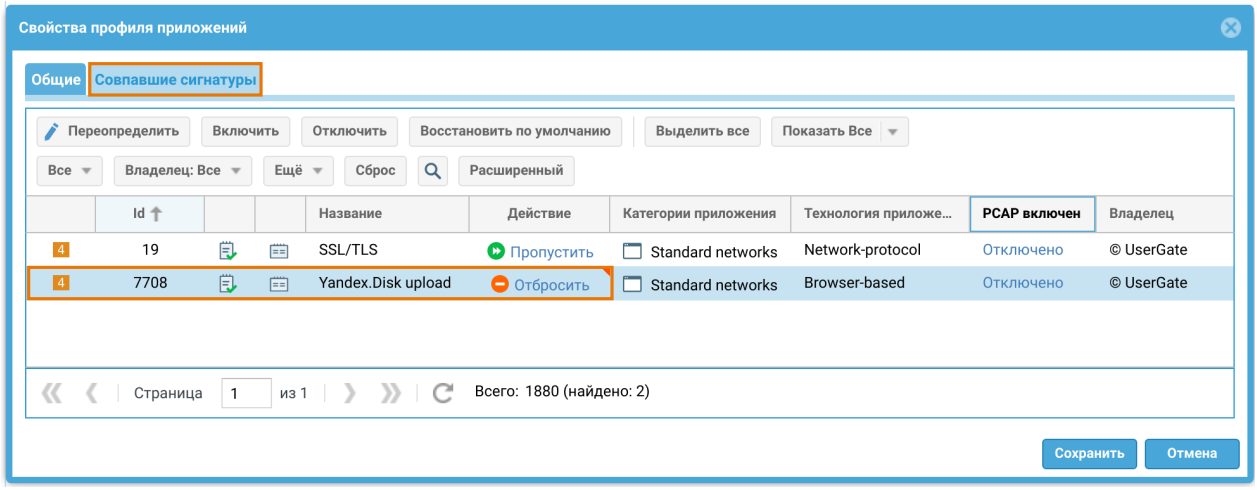


It allows access and uploading files to Yandex Disk, and the rest of traffic is marked as non-identified and rejected by the SSL/TLS signature or by the signatures for non-identified applications.

Example 3. Blacklist profile with associated signatures

In this example all traffic is allowed except for the traffic matching the Yandex.Disk upload signature (id=7708). For this case we need to take into account the dependency of the Yandex. Disk signature from the SSL/TLS (id=19) protocol signature. The relation to the Yandex.Disk and Yandex Services signatures **is not taken into account in case of blocking**. The application profile in this example will look as follows:





It denies uploading files to Yandex Disk, and the rest of traffic is allowed by the SSL/TLS signature or by the signatures for non-identified applications.

Using application profiles

The administrator can create any number of profiles. It is recommended to limit the number of signatures in the profile only to those that are necessary for protecting a certain service. A large number of signatures increases the traffic processing time and CPU load.

The applications profile is applied in the allow rule of the firewall.

The firewall rules are processed from top to bottom and the session matches the first rule which meets all the conditions of the rule (source/destination addresses/

zones, users etc.). After matching the allow rule with the application profile the traffic is analyzed using the profile signatures. Both forward and reverse packets are analyzed in accordance with the filter conditions, regardless of where the connection is established from. If the profile signatures are matched, the action specified in the profile filters will be performed, and the record will be added to the Traffic log, if logging is enabled. If none of the signatures were matched, the action specified in the profile for the non-identified traffic will be applied to the traffic.

If a signature with the Block IP action is triggered, then the source or destination IP address (depending on the setting) is blocked for the time specified in the settings. The IP addresses blocked by signatures are shown on the **Diagnostics and monitoring** page of the **IDPS/L7 blocked IP addresses** section (for more details please see the IDPS/L7 blocked IP addresses section).

Application Groups

Here the user can manage (create, update, and delete) application groups. Application groups can be used to configure traffic shaping rules.

To create an application group:

Name	Description
Step 1. Create a group.	In the **Libraries → Application Groups** section, click **Add**, specify the name and, optionally, a description of the application group.
Step 2. Add application signatures to a group.	In the **Applications** pane, click **Add** and select application signatures to add to the group. To add all application signatures, use the **Add all** button.

Emails

The **Emails** library item allows you to create email groups that can later be used in email traffic filtering rules and notifications.

To add a new email group, follow these steps:

Name	Description
Step 1. Create an email group.	In the **Email groups** pane, click **Add** and give a name to the new group.
Step 2. Add emails to the group.	Highlight the group just created, click **Add** in the **Emails** pane, and add the desired email addresses.

The administrator can create custom email lists and distribute them centrally to all UserGate firewalls. To create such a list, follow these steps:

Name	Description
Step 1. Create a file with the relevant email list.	Create a file named **list.txt** with the email list.
Step 2. Create an archive containing this file.	Put the file in a ZIP archive named **list.zip**.
Step 3. Create a version file for the list.	Create a file named **version.txt** and specify the database version number inside it, such as 3. On each update of the morphological dictionary, the version number must be incremented.
Step 4. Upload the files to a web server.	Upload the **list.zip** and **version.txt** files to your website so that they can be downloaded.
Step 5. Create an email list and specify an update URL for it.	On each DCFW, create an address list. When creating the list, select **Updatable** as the list type and enter the address for downloading updates. DCFW will check for a new version on your website according to the configured updates download schedule. The schedule can be configured in the list properties. The available options are: • Disabled: update checking will not be performed for the selected item • Daily • Weekly • Monthly • Every ... hours • Every ... minutes • Advanced. With the Advanced option, a crontab-like format is used where the date/time string consists of six space-separated fields. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6,

Name	Description
	where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

Phones

The **Phones** library items allows you to create phone groups that can later be used in SMPP notification rules.

To add a new phone group, follow these steps:

Name	Description
Step 1. Create a phone group.	In the **Phone groups** pane, click **Add** and give a name to the new group.
Step 2. Add phone numbers to the group.	Highlight the group just created, click **Add** in the **Phones** pane, and add the desired phone numbers.

The administrator can create custom phone lists and distribute them centrally to all UserGate firewalls. To create such a list, follow these steps:

Name	Description
Step 1. Create a file with the relevant phone list.	Create a file named **list.txt** with the phone list.
Step 2. Create an archive containing this file.	Put the file in a ZIP archive named **list.zip**.
Step 3. Create a version file for the list.	Create a file named **version.txt** and specify the database version number inside it, such as 3. On each update of the

Name	Description
	morphological dictionary, the version number must be incremented.
Step 4. Upload the files to a web server.	Upload the **list.zip** and **version.txt** files to your website so that they can be downloaded.
Step 5. Create a phone list and specify an update URL for it.	On each DCFW, create an address list. When creating the list, select **Updatable** as the list type and enter the address for downloading updates. DCFW will check for a new version on your website according to the configured updates download schedule. The schedule can be configured in the list properties. The available options are: • Disabled: update checking will not be performed for the selected item • Daily • Weekly • Monthly • Every ... hours • Every ... minutes • Advanced. With the Advanced option, a crontab-like format is used where the date/time string consists of six space-separated fields. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6, where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

IDPS Signatures

IDPS signatures are a set of strings (patterns) and semantic expressions (filters, modifiers, other constructions) that allow identifying/marking a network attack and taking certain actions. Signatures are added to IDPS profiles and used in firewall rules to detect intrusions and protect the network.

UserGate can use two types of IDPS signatures:

- Proprietary signatures
- Customized user signatures

Proprietary IDPS signatures are created by UserGate developers and automatically added to the system library if there is an appropriate license. In the list of signatures in the library, such signatures are marked as @UserGate in the **Owner** column. In proprietary signatures, the user can reconfigure the following parameters:

- **Enabling/disabling** the signature.
- **Signature logging**.
- **Recording in pcap file** if the signature is triggered.
- The **action** to take, in case a signature was triggered (i.e., found in the traffic). Possible actions: pass packet, drop packet, drop packet with TCP connection break, block source and/or destination IP address.

Additional optional proprietary signature parameters:

- **Apply to**: this one needs to be configured, if the **Action** parameter takes **Drop the packet and close TCP connection** and **Block the source and/or destination IP address** values. The parameter has the following values: **Source**, **Destination**, **Both**.
- **Duration**: can be configured if the **Action** parameter has the **Block the source and/or destination IP address** value. The number that specifies block time for an IP address.

Some signatures also allow the following additional settings to be configured:

- **Direction**: configuring aggregation by a source or destination IP address.
- **Triggering frequency**: the number of triggered alerts per time unit (in seconds).

Examples of signatures with these settings:

1064, 1672, 1711, 1732, 1738, 1740, 1741, 5315, 5611, 5612, 5657, 5699, 5701, 5757,

13003, 17002, 17003, 17004, 45022, 3029251, 3031043, 3031817, 3032798, 3032823,

3033202, 3033935, 3034466, 3035136, 3037395, 3037608, 3037708, 3037771, 3039602,

3039703, 3039876, 3039883, 3040088, 3040443, 3042187, 3046218, 3046453, 3046609,

3049480, 3050453, 3050940, 3051410, 3051613, 3051634, 90000000, 90000001, 90000002,

90000003, 90000004, 90000005, 90000006, 90000007, 90000008, 90000009, 90000010

Note

This list contains examples of signatures that have the additional configuration parameters *Triggering frequency* and *Direction*. As the product develops, the number of such signatures will increase.

After changing the default parameter settings for proprietary signatures, the **Status** column will show **Changed**. User-changed settings of proprietary IDPS signatures can be returned to their original state. To do this, in the administrator web console, in the **Libraries → IDPS Signatures** section, select the signature in the list and click the **Restore default** button.

Custom IDPS signatures are created by the user.

To create a customized IDPS signature, in the administrator web console, go to the **Libraries → IDPS signatures** section and click the **Add** button. Then fill in the fields with the signature parameters. Network vulnerability indicators are described using the UASL syntax (UserGate Application and Security Language).

When creating a customized signature, fill in the following fields:

Name	Description
Enabled	Signature on/off indicator.
Id	Signature ID. If the field is left empty, a free ID from the user pool will be issued.
Name	The name of the signature.

Name	Description
Description	Signature description.
Threat level	Threat level defined by the signature. The following values are defined: • **1**: very low • **2**: low • **3**: medium • **4**: high • **5**: very high
Class type	The signature class determines the attack type that is detected using this signature. In addition, it determines the general events that are not related o the attack but can be relevant in certain cases; e.g., detecting the establishment of a TCP session. The class list (can be extended): • **arbitrary-code-execution**: attempt to run arbitrary code • **attempted-admin**: attempt to obtain administrative privileges • **attempted-dos**: attempt to launch a Denial-of-Service (DoS) attack • **attempted-recon**: attempt to launch an attack aimed at leaking data • **attempted-user**: attempt to obtain user privileges • **bad-unknown**: potentially unwanted traffic • **buffer overflow**: attempt to launch a buffer-overflow attack • **command-and-control**: attempt to communicate with a C&C center • **default-login-attempt**: attempt to log in with the default username/password • **denial-of-service**: Denial-of-Service attack detected • **exploit-kit**: exploit kit detected • **information disclosure**: data leak • **memory corruption**: attempt to launch a memory corruption attack • **misc-activity**: other activity • **misc-attack**: attack detected • **network-scan**: network scanning • **path traversal**: attempt to launch an attack that works by traversing file paths on the server where the application is running

Name	Description
	• **policy-violation**: network policy violation • **protocol-command-decode**: unusual protocol command detected • **shellcode-detect**: shell code detected • **string-detect**: suspicious string detected • **successful-recon-limited**: information leak • **suspicious-login**: attempt to log in using a suspicious username • **system-call-detect**: attempt to invoke system calls • **targeted-activity**: targeted activity detected • **trojan-activity**: network Trojan detected • **uncaught exception**: exception not handled by the application.
Category	A signature category is a group of signatures that have common parameters. The list of categories (can be extended): • **adware pup**: unwanted adware • **attack_response**: signatures that specify responses to known network attacks • **bruteforce**: brute-force attack • **coinminer**: downloading, installation, and runtime activity of known miners • **dns**: known DNS vulnerabilities • **dos**: known signatures of denial-of-service (DoS) attacks • **exploit**: signatures of known exploits • **ftp**: known FTP vulnerabilities • **icmp**: known ICMP protocol vulnerabilities • **imap**: known IMAP vulnerabilities • **info**: potential data leaks • **ldap**: known LDAP vulnerabilities • **malware**: downloading, installation, and runtime activity of known malware • **misc**: other known signatures • **netbios**: known NetBIOS protocol vulnerabilities • **p2p**: peer-to-peer traffic detected • **phishing**: signatures of known phishing attacks • **policy**: cybersecurity policy violation • **pop3**: known POP3 protocol vulnerabilities • **rpc**: known RPC protocol vulnerabilities • **scada**: known SCADA protocol vulnerabilities

Name	Description
	• **scan**: signatures of attempts to scan the network for known applications • **shellcode**: signatures specifying known attempts at launching shells • **sip**: known SIP protocol vulnerabilities • **smb**: known SMB protocol vulnerabilities • **smtp**: known SMTP protocol vulnerabilities • **snmp**: known SNMP protocol vulnerabilities • **sql**: known SQL vulnerabilities • **telnet**: known attempts at cracking via the telnet protocol • **tftp**: known TFTP protocol vulnerabilities • **user_agents**: signatures of suspicious Useragents • **voip**: known VoIP protocol vulnerabilities • **web_client**: signatures of known attempts at cracking various web clients, such as Adobe Flash Player • **web_server**: signatures specifying known attempts at cracking various web servers • **web_specific_apps**: signatures specifying known attempts at cracking various web applications • **worm**: signatures specifying network activity of known network worms
Signature operating system	The operating system for which this signature is developed. • **Windows** • **Linux** • **BSD** • **Mac OS** • **Solaris** • **Cisco** • **IOS** • **Android** • **Other**
CVE	Vulnerability ID according to the CVE registry.
BDU	Vulnerability ID according to the BDU registry.
URL	Optional link to a resource with the description of the vulnerability.

Name	Description
UASL	Description of the signature features using the UASL syntax.
General Settings	• **Action**: the response to signature detection. The following values are defined: ◦ **None**: no action defined ◦ **Pass**: allow the packet ◦ **Drop**: drop the packet ◦ **Reset**: drop the packet and abort the TCP connection (send a TCP reset) ◦ **Block IP**: block the source and/or destination IP address • **Log**: ◦ **Enable**: enable event logging ◦ **Disable**: disable event logging • **PCAP file**: trace the signature detection and write the results in a PCAP file ◦ **Enable**: enable tracing ◦ **Disable**: disable tracing • **Apply to**: what the **Reset** or **Block IP** actions should apply to. The available options are: ◦ **Source**: the **Reset** or **Block IP** action is applied to the source IP address of the packet ◦ **Destination**: the **Reset** or **Block IP** action is applied to the destination IP address of the packet ◦ **Both**: the **Reset** or **Block IP** action is applied to both the source and destination IP addresses of the packet • **Duration**: the block duration for the **Block IP** action

IDPS Profiles

Purpose of the IDPS Profile

The IDPS profile allows you to create a dynamic set of IDPS signatures designed to detect intrusions and protect certain services. The profile is dynamic because the profile does not explicitly contain any signatures, but contains filters that collect a set of signatures. Both descriptive signature fields and settings are used for filtering.

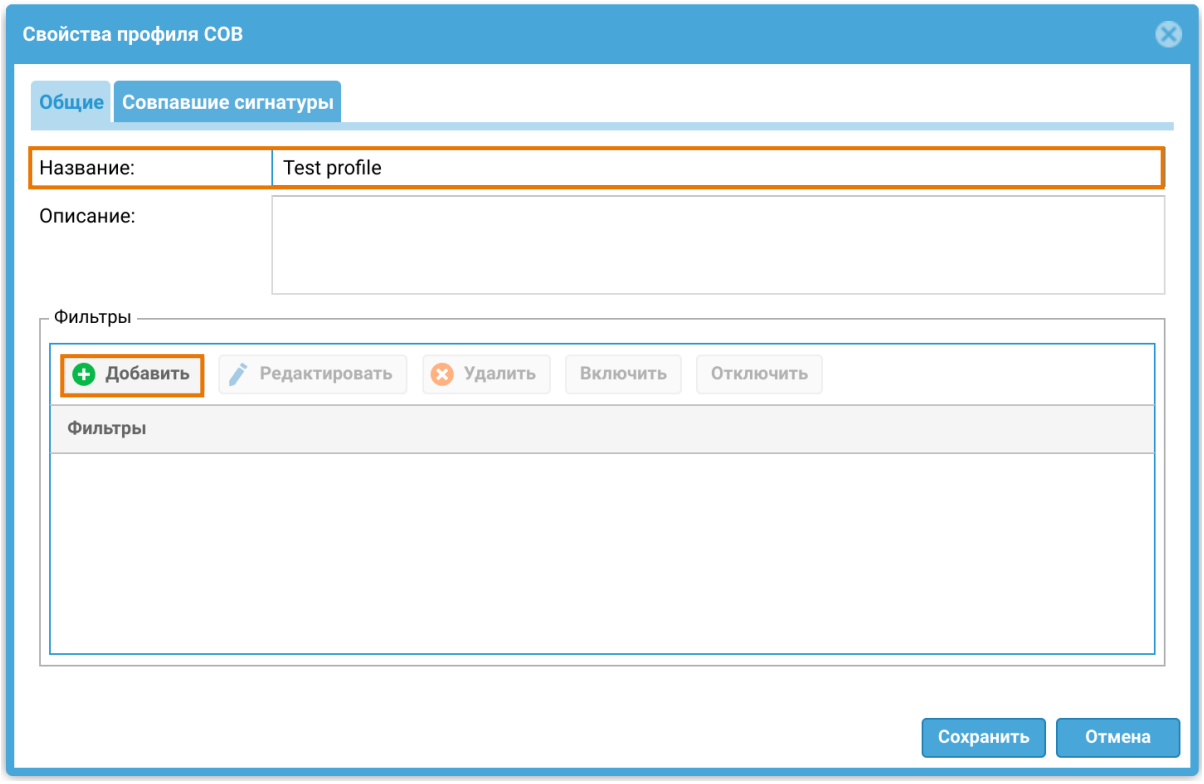
As a result, when the signature library changes, the profiles will dynamically collect new sets of signatures that satisfy the profile filters.

In addition to creating the required set of signatures, the profile can define actions that will be performed on traffic filtered by the signatures.

Creating an IDPS Profile in the Administrator Web Console

In the administrator web console, go to the **Libraries → IDPS profiles** section.

Click **Add** and fill in the corresponding fields in the profile properties:



1. In the **Name** field, specify the name of the profile being created.

2. In the **Description** field, optionally specify the purpose of the profile.

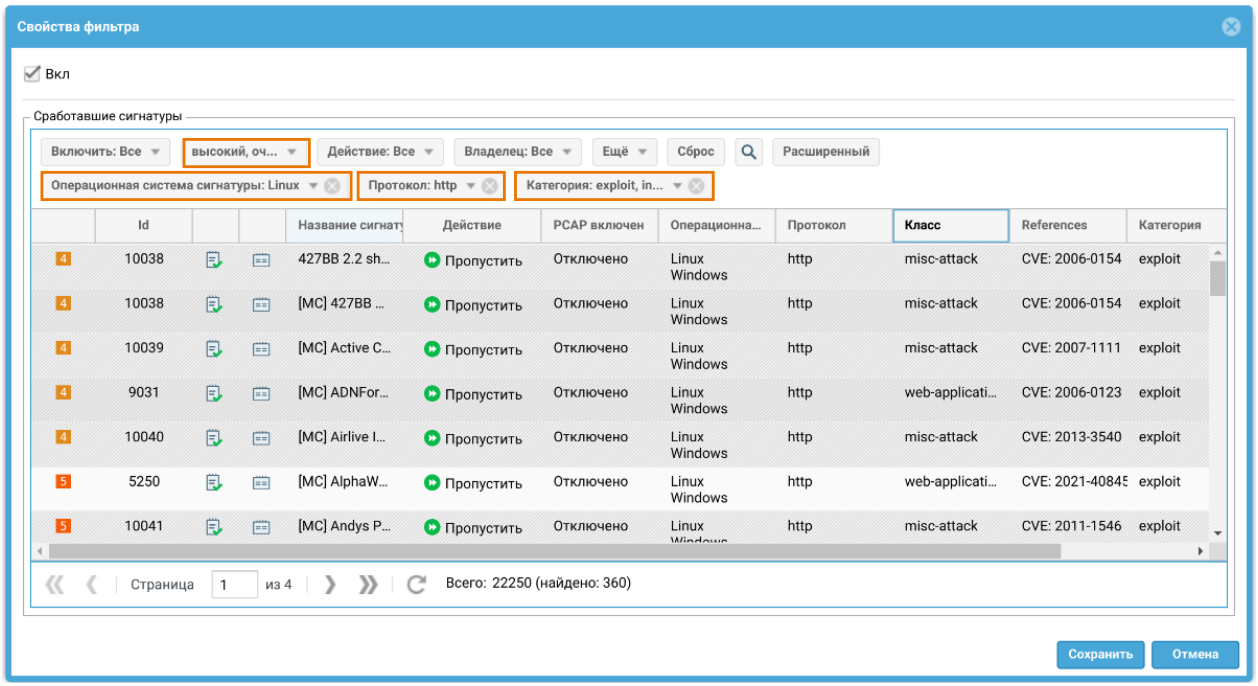
3. In the **Filters** area, add filters to select the required signatures from the library.

4. The **Matched signatures** tab displays a preview of the IDPS signatures selected by all the profile filters and the configured actions to be performed on traffic filtered by these signatures.

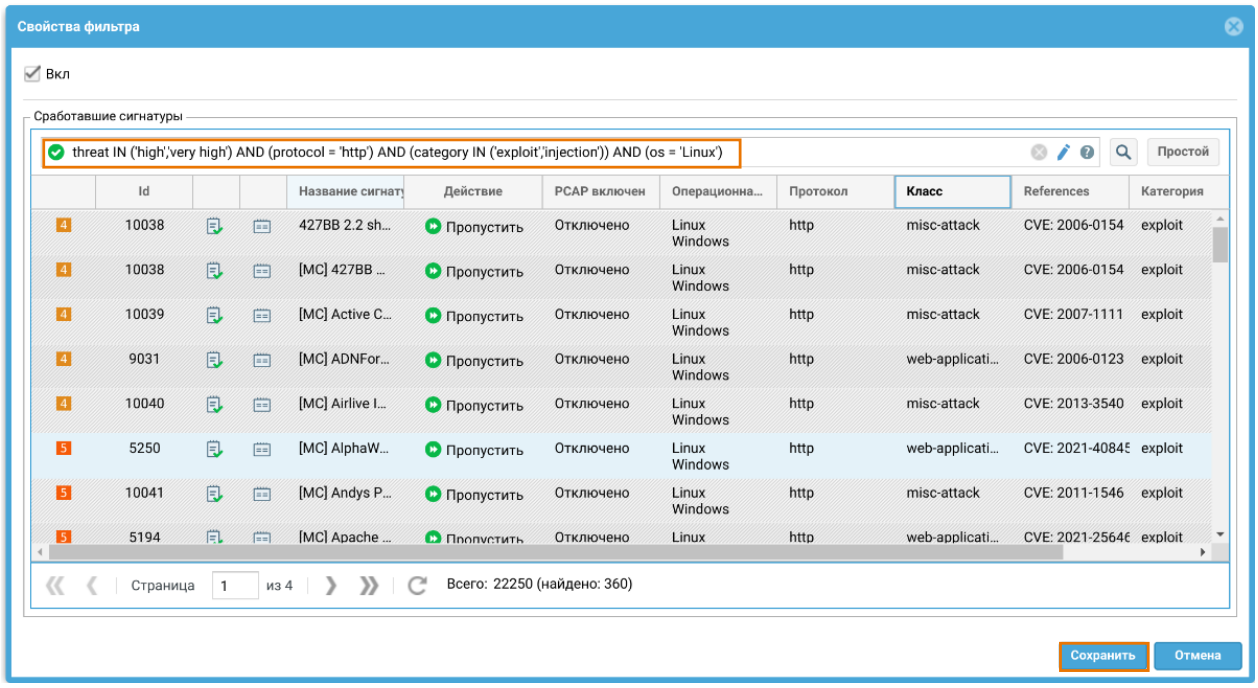
Configuring Signature Filters in the IDPS Profile

To create a signature filter, click the **Add** button in the **Filters** area. The filter properties window will open.

You can create a filter by selecting options in the toolbar. The window below the toolbar will display the signatures selected by this filter:



You can also create a filter by describing it using SQL-like syntax. To do this, click **Advanced** on the toolbar and describe the filter selection properties in the line that opens:



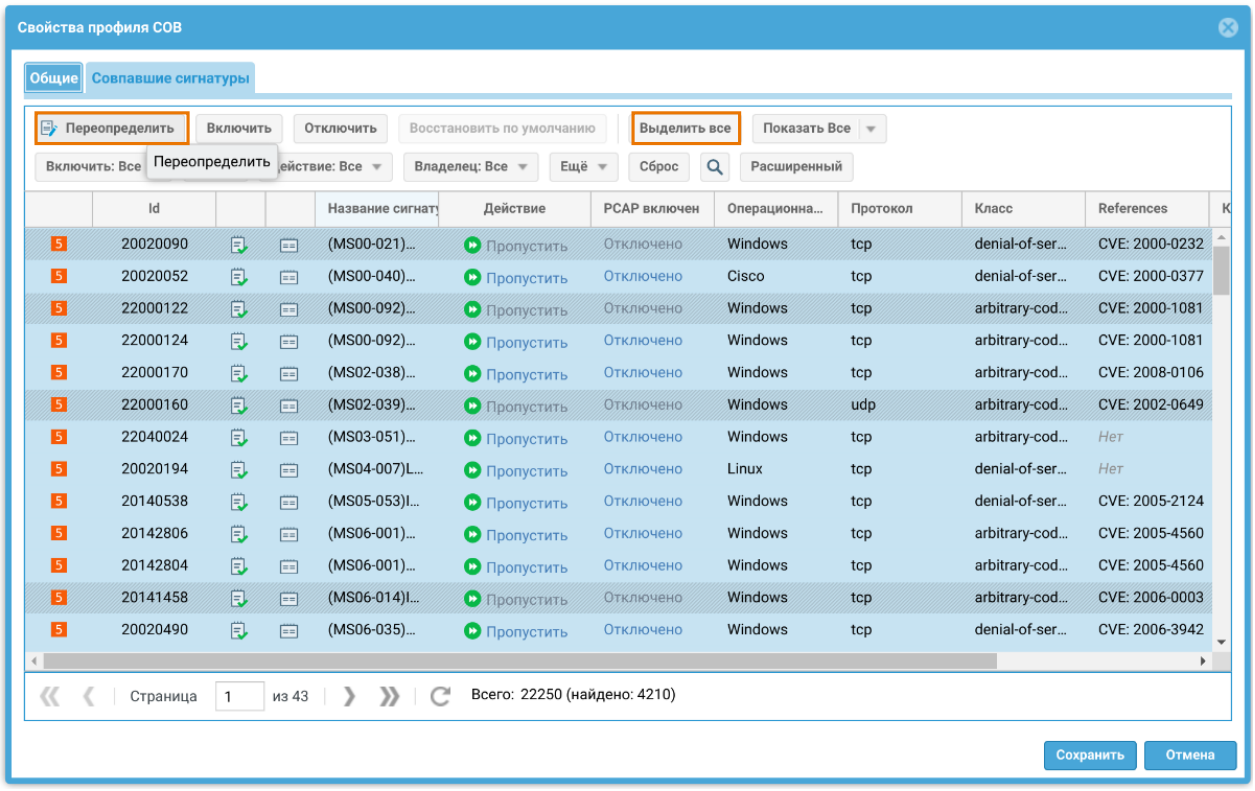
To save the created filter, click the **Save** button.

You can use several filters at once in one profile.

Filters in a profile use the logical OR. For example, if two filters "category = injection" and "threat = low" are added to a profile, they are equivalent to a filter "category = injection OR threat = low".

Configuring Signature Parameters in the IDPS Profile

Within the profile, you can override signature parameters such as action, logging, writing to a pcap file, enabling/disabling a signature. To do this, select the desired signatures in the list of matching signatures and click the **Override** button on the toolbar.



Changing the signature settings in the IDPS profile has a higher priority than the settings of the same signatures on the IDPS signatures page. Changed IDPS signature settings can be returned to their original state by selecting a signature in the profile signature list and clicking the **Restore default** button in the profile toolbar.

Applying IDPS Profiles

The administrator can create any number of profiles. It is recommended to limit the number of signatures in the profile only to those that are necessary for protecting a certain service. A large number of signatures increases the traffic processing time and CPU load.

The IDPS profile is applied in an allow firewall rule.

Firewall rules are processed from top to bottom, and a session is matched by the first rule that meets all the conditions. Once the traffic is matched with the rule with an IDPS profile, it gets analyzed using the set of signatures defined in this profile. Both forward and return packets are analyzed according to the filter conditions, regardless of the connection's origin. When profile signatures are triggered, the action configured in the profile is executed and a corresponding entry is registered in the IDPS log, provided logging is enabled. If none of the profile signatures are found, the traffic is passed.

If a signature with the Block IP action is triggered, then the source or destination IP address (depending on the setting) is blocked for the time specified in the settings.

The IP addresses blocked by signatures are shown on the **Diagnostics and monitoring** page of the **IDPS/L7 blocked IP addresses** section (for more details please see the IDPS/L7 blocked IP addresses section).

Notification Profiles

A notification profile defines a transport that can be used to deliver notifications to the users. Two types of transport are supported:

- SMTP for delivering messages by email.
- SMPP for message delivery by SMS via virtually any cellular provider or the numerous SMS distribution centres.

To create an SMTP notification profile, go to the **Libraries → Notification profiles** section, click **Add**, select the **Add SMTP notification profile** option, and fill in the relevant fields:

Name	Description
Name	Profile name.
Description	Profile description.
Host	The IP address or FQDN of the SMTP server that will be used for sending emails.
Port	The TCP port used by the SMTP server. Usually, SMTP uses port 25, and SMTP with SSL uses port 465. Consult your email server administrator regarding this value.
Connection security	The following outgoing email security options are available: None, STARTTLS, and SSL.
Authorization	Turns on authorization for SMTP server connection.
Login name	The account name for connecting to the SMTP server.
Password	The account password for connecting to the SMTP server.

To create an SMPP notification profile, go to the **Libraries → Notification profiles** section, click **Add**, select the **Add SMPP notification profile** option, and fill in the relevant fields:

Name	Description
Name	Profile name.
Description	Profile description.
Host	The IP address or FQDN of the SMPP server that will be used for sending SMS messages.
Port	The TCP port used by the SMPP server. Usually, SMPP uses port 2775, and SMPP with SSL uses port 3550.
SSL	Specifies whether or not SSL encryption is used.
Login name	The account name for connecting to the SMPP server.
Password	The account password for connecting to the SMPP server.
Phone translation rules	In certain cases, the SMPP provider expects a phone number in a specific format, such as 0123456789. To meet the provider's requirements, you can configure the replacement of the leading phone number digits with others. For example, you can replace the leading +971 with 0.

NetFlow Profiles

NetFlow is a network traffic accounting protocol developed by Cisco Systems and currently supported by numerous vendors. To collect traffic information using NetFlow, the following components are required:

- Sensor: gathers statistics on the traffic passing through it and sends this data to the collector.
- Collector: receives the data from the sensor and stores it.
- Analyzer: analyzes the data gathered by the collector and forms human-readable reports (often in the form of graphs or charts).

DCFW can function as a sensor. To collect and send out statistics on the traffic passing through a specific DCFW network interface, follow these steps:

1. Create a new NetFlow profile.
2. Assign the newly created NetFlow profile to the network interface on which statistics are to be collected.

To create an NetFlow profile, go to the **Libraries → NetFlow profiles** section, click **Add**, and provide the desired settings:

Name	Description
Name	Netflow profile name.
Description	A description of the NetFlow profile.
NetFlow collector IP address	The IP address of the server where the sensor will send the statistics.
NetFlow collector port	The UDP port on which the collector will receive the statistics.
Netflow protocol version	The NetFlow protocol version to be used. The protocol version must match on the sensor and collector.
Active flow timeout, (sec.)	In case of long data flows, such as transmitting a large file over the network, the time after which statistics will be sent to the collector without waiting for the flow to be completed. The default value is 1800 seconds.
Inactive flow timeout, (sec.)	The time reserved for completing an inactive flow. The default value is 15 seconds.
Maximum flows	Maximum number of counted flows from which statistics are gathered and sent. This limit is required to protect against DoS attacks. After reaching this number of flows, any subsequent flows will be ignored. The default value is 2000000. To remove the limit, set this to 0.
Send NAT information	Send information on network address translation as part of NetFlow statistics.
Template refresh rate (packets)	The number of packets after which the template is sent to the receiving host (only for NetFlow 9/10). The template contains information about the configuration of the device and various statistical information. The default value is 20 packets.
Period to re-send old template (sec.)	The time interval after which the old template is sent to the receiving host (only for NetFlow 9/10). The template contains information about the configuration of the device and various statistical information. The default value is 1800 seconds.

LLDP Profiles

Link Layer Discovery Protocol (LLDP) is a link layer protocol that allows network devices operating in a local network to advertise themselves, report their own characteristics, and receive similar information from others. The information collected as part of LLDP operation is stored in the device.

To create a security profile, go to the **Libraries → LLDP profiles** section, click **Add**, and provide the following settings:

Name	Description
Name	The name of the LLDP profile.
Description	A description of the LLDP profile.
Port status	Mode: • **RX and TX**: NGFW will send LLDP information and analyze LLDP information received from the neighbors. • **RX only**: NGFW will not send LLDP information but will analyze LLDP information received from the neighbors. • **TX only**: NGFW will send LLDP information but reject LLDP information received from the neighbors.

SSL Profiles

Using an SSL profile, you can specify SSL protocols or individual encryption and digital signature algorithms that can later be used in web console, auth page, and block page settings.

To create an SSL profile, go to the **Libraries → SSL profiles** section, click **Add**, and provide the desired settings:

Name	Description
Name	The name of the SSL profile.
Description	A description of the SSL profile.
SSL protocols	**Min TLS version**: the minimum TLS version that can be used with this profile. **Max TLS version**: the maximum TLS version that can be used with this profile.

Name	Description
	These two settings determine the TLS version range that will be supported by this profile.
Ciphers suites	In this section, you can choose the desired encryption and digital signature algorithms. The enumerated options are presented as strings listing the specific algorithm pairs. The administrator may choose to select only those algorithm pairs that they deem necessary for the secure operation of the organization. The supported combinations are: • TLS AES 128 CCM SHA256 • TLS AES 128 GCM SHA256 • TLS AES 256 GCM SHA384 • TLS DHE DSS with 3DES EDE CBC SHA • TLS DHE DSS with AES 128 CBC SHA • TLS DHE DSS with AES 128 CBC SHA256 • TLS DHE DSS with AES 128 GCM SHA256 • TLS DHE DSS with AES 256 CBC SHA • TLS DHE DSS with AES 256 CBC SHA256 • TLS DHE DSS with AES 256 GCM SHA384 • TLS DHE RSA with 3DES EDE CBC SHA • TLS DHE RSA with AES 128 CBC SHA • TLS DHE RSA with AES 128 CBC SHA256 • TLS DHE RSA with AES 128 GCM SHA256 • TLS DHE RSA with AES 256 CBC SHA • TLS DHE RSA with AES 256 CBC SHA256 • TLS DHE RSA with AES 256 GCM SHA384 • TLS ECDH ECDSA with 3DES EDE CBC SHA • TLS ECDH ECDSA with AES 128 CBC SHA • TLS ECDH ECDSA with AES 128 CBC SHA256 • TLS ECDH ECDSA with AES 128 GCM SHA256 • TLS ECDH ECDSA with AES 256 CBC SHA • TLS ECDH ECDSA with AES 256 CBC SHA384 • TLS ECDH ECDSA with AES 256 GCM SHA384 • TLS ECDH RSA with 3DES EDE CBC SHA • TLS ECDH RSA with AES 128 CBC SHA • TLS ECDH RSA with AES 128 CBC SHA256 • TLS ECDH RSA with AES 128 GCM SHA256 • TLS ECDH RSA with AES 256 CBC SHA • TLS ECDH RSA with AES 256 CBC SHA384 • TLS ECDH RSA with AES 256 GCM SHA384

Name	Description
	• TLS ECDHE ECDSA with 3DES EDE CBC SHA • TLS ECDHE ECDSA with AES 128 CBC SHA • TLS ECDHE ECDSA with AES 128 CBC SHA256 • TLS ECDHE ECDSA with AES 128 GCM SHA256 • TLS ECDHE ECDSA with AES 256 CBC SHA • TLS ECDHE ECDSA with AES 256 CBC SHA384 • TLS ECDHE ECDSA with AES 256 GCM SHA384 • TLS ECDHE RSA with 3DES EDE CBC SHA • TLS ECDHE RSA with AES 128 CBC SHA • TLS ECDHE RSA with AES 128 CBC SHA256 • TLS ECDHE RSA with AES 128 GCM SHA256 • TLS ECDHE RSA with AES 256 CBC SHA • TLS ECDHE RSA with AES 256 CBC SHA384 • TLS ECDHE RSA with AES 256 GCM SHA384 • TLS GOST2012256 with 28147 CNT IMIT • TLS GOSTR341001 with 28147 CNT IMIT • TLS RSA with 3DES EDE CBC SHA • TLS RSA with AES 128 CBC SHA • TLS RSA with AES 128 CBC SHA256 • TLS RSA with AES 128 GCM SHA256 • TLS RSA with AES 256 CBC SHA • TLS RSA with AES 256 CBC SHA256 • TLS RSA with AES 256 GCM SHA384
Set encryption algorithms for standard protocols	You can use this section to facilitate the selection of encryption and digital signature algorithms for standard TLS protocols. The administrator can specify the desired TLS protocol version in the **Select protocol and set ciphers set** field and click **Apply**, after which the algorithms that match the selected protocol versions will be automatically selected. You can repeat the process to add multiple TLS protocol versions.

There are several default SSL profiles in the product that can be used by the administrator as is or edited/deleted if necessary. The following predefined SSL profiles exist:

Name	Description
Default SSL profile	Contains encryption and digital signature algorithms supported by TLS v1.1 to TLS v1.2. These are the most common protocol

Name	Description
	versions currently used in the Internet. This profile is used by default for: • Captive portal auth page • Block page
Default SSL profile (TLSv1.3)	Contains encryption and digital signature algorithms supported by TLS v1.3. Not used by default.
Default SSL profile	Can be used in organizations where the use of such algorithms is required. The browsers used must also support these protocols. Not used by default.
Default SSL profile (web console)	Contains encryption and digital signature algorithms supported by TLS v1.0 to TLS v1.2. This profile is used by default to provide SSL access to the web console. **Important!** Use caution when editing this profile. Specifying algorithms not supported by your browser can cause loss of access to the web console!

BFD Profiles

BFD (Bidirectional Forwarding Detection) is a protocol that operates at the interface and routing protocol levels and is designed to quickly detect failures between two neighboring routers, including interfaces, data links, and forwarding mechanisms. BFD operates over any data transmission protocol (network layer, link layer, tunnels, etc.) used between the two systems. BFD packets are transmitted as the payload of an encapsulating protocol suitable for the specific environment and network. BFD can operate at several levels of the system.

BFD routes exchange packets at a negotiated rate. If there are no incoming packets from a BFD-supporting router, that router is considered inoperative. BFD shares this information with the correspondent routing protocols, and the routing information is updated. BFD helps detect a single-end device failure and is used for the quick convergence of routing protocols.

A BFD profile is a configuration or a set of parameters used in dynamic routing protocols (BGP, OSPF) to define how the bidirectional forwarding detection functionality should work. Typically, a BFD profile includes parameters such as the desired detection time, hold time, and other parameters that determine the link fault detection speed and the response time of network devices in case of a fault.

Configuring and using profiles ensures prompt network fault detection, which helps speed up traffic rerouting to different interfaces and boost network reliability.

Configuring BFD for OSPF allows the corresponding BFD session connection events to instantly update the OSPF interface status.

In the case of BGP, BFD can also be used to regulate the failure detection time. Configuring BFD to detect link failures more quickly allows for faster response and improved BGP routing convergence.

To create a BFD profile, go to **Libraries → BFD profiles**, click **Add**, and specify the desired settings:

Name	Description
Name	Set the BFD profile name.
Detect multiplier	Determine the detection time multiplier. The local system calculates the connection fault detection time as the product of a multiplier of the detection time received from the remote system and the agreed transmission interval of the remote system. If the BFD does not receive the control packet before the detection time expires, then the connection is considered to have failed. For example, if the transmission interval is 300 ms and the multiplier is 3, then the local system will detect failures only after 900 ms of no packets being received.
Receive interval	Configure the interval for receiving BFD control packets (the minimum time required between packets). The interval is not consistent between nodes. To determine the interval, each node compares its transmission interval with the reception interval of its neighbor - the larger of the two values is accepted as the transmission interval for this node. The default value is 50ms.
Transmit interval	Specify the transmission interval of BFD control packets; the interval must be consistent between nodes. The default value is 50ms.
Echo receive interval	The minimum time interval after which the system will be able to receive echo packets. The default value is 50ms.
Echo transmit interval	Configure the minimum transmission interval at which this system will be able to send BFD echo packets. The default value is 50ms.

Name	Description
Echo mode	Enable/disable **Echo mode** for data transmission. This mode is off by default. When the Echo feature is active, the BFD Echo packet stream is sent to the remote system, which returns them back along the same forwarding path. If a certain number of packets from the echo stream has not been received, the session is considered inoperative. The advantage of **Echo mode** is that it only tests the forwarding path on the remote system. This allows you to reduce the delay when passing the route and reduce the time spent detecting failures. The echo mode is not supported in multihop networks (see RFC 5883).
Passive mode	Enable or disable the **Passive** mode. When operating in the **Passive** mode, the system waits for control packets from neighbors and responds to them if they are received. This feature is useful when the router is central in a star-topology network, and you want to avoid unnecessary BFD control packets. By default, the **Active** mode is used. When operating in the **Active** mode, the node sends control packets to the neighboring node. **Important!** Both nodes cannot operate in the Passive mode. At least one of them (or both) must work in the Active mode.
Minimum-ttl	For multi-hop sessions only: configure the minimum lifetime value (number of hops) that BFD will accept in the BFD control packet. Can take values from 1 to 254. All packets with a lower TTL value will be discarded. Setting this value is necessary to set more stringent packet checking requirements to avoid receiving BFD control packets from other sessions. The default value is 254 (meaning that we expect only a single hop between the system and its peer).

UserID Agent Syslog Filters

When using syslog as an event source, UserGate filters events according to the agent's UserID filters specified by syslog. Syslog filters are standard regular

expressions that users can write themselves. Three types of filters are provided as standard:

Name	Description
SSH Authentication	A filter for tracking SSH login/logout events in syslog logs.
Unix PAM Authentication	A filter to track user logon/logoff events using **Pluggable Authentication Modules** (PAM) technology in syslog logs.
UserGate WEC Agent	A filter designed to track events transmitted via syslog from the UserID agent for AD/WEC servers. (Available starting from version 7.2.0)

> **Note**
> **You can create additional rules using regular expressions. Thus, syslog filters are a versatile tool that can be used in almost any case.**

The found events are displayed on the **Logs and reports** tab, under **Logs → User-ID agent →** <0>Syslog.

Scenarios

The Purpose of Scenarios

With UserGate DCFW, the attack detection to response time can be reduced considerably thanks to a concept called SOAR (Security Orchestration, Automation, and Response). DCFW implements this concept using a scenario-based mechanism. A scenario is an additional condition in the firewall, traffic shaping, content filtering, PBR, and DoS protection rules that allows an administrator to configure DCFW's response to certain events that have occurred within a prolonged time frame.

An example of a scenario work could be a task to temporarily apply traffic shaping to a user who has selected a set traffic limit.

Configuring Scenarios

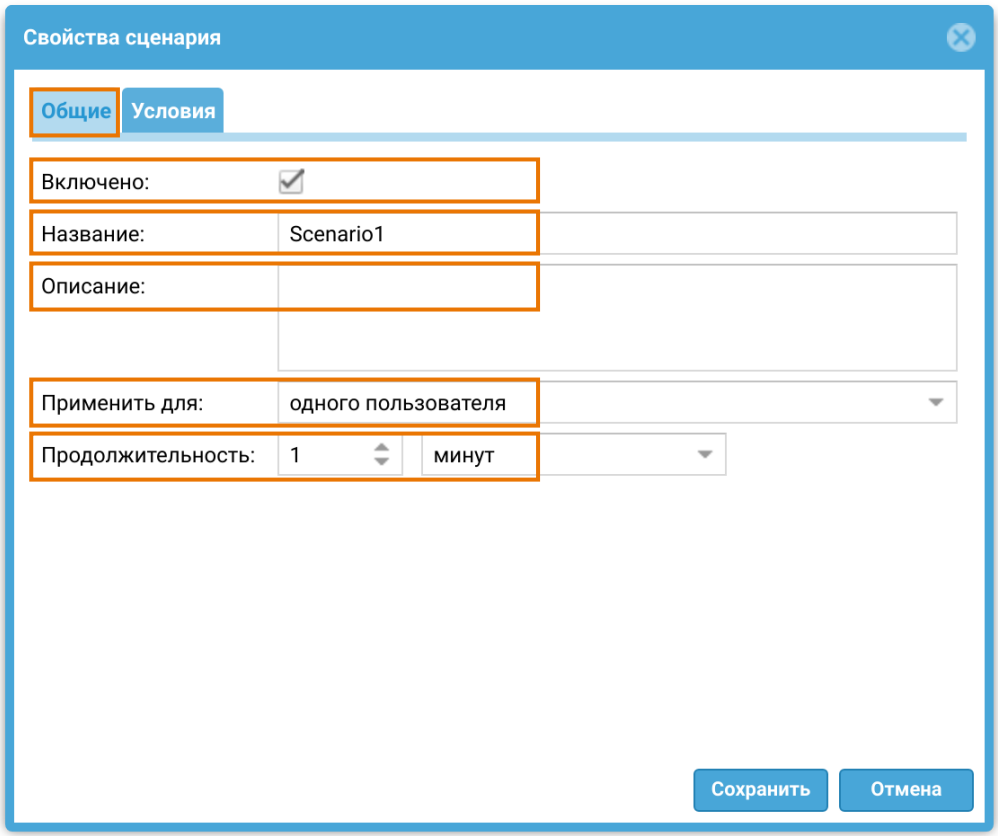
To get started with scenarios, follow these steps:

1. Create a scenario.

2. Apply the created scenario in the firewall rules, bandwidth rules, content filtering rules, PBR rules, DoS protection rules.

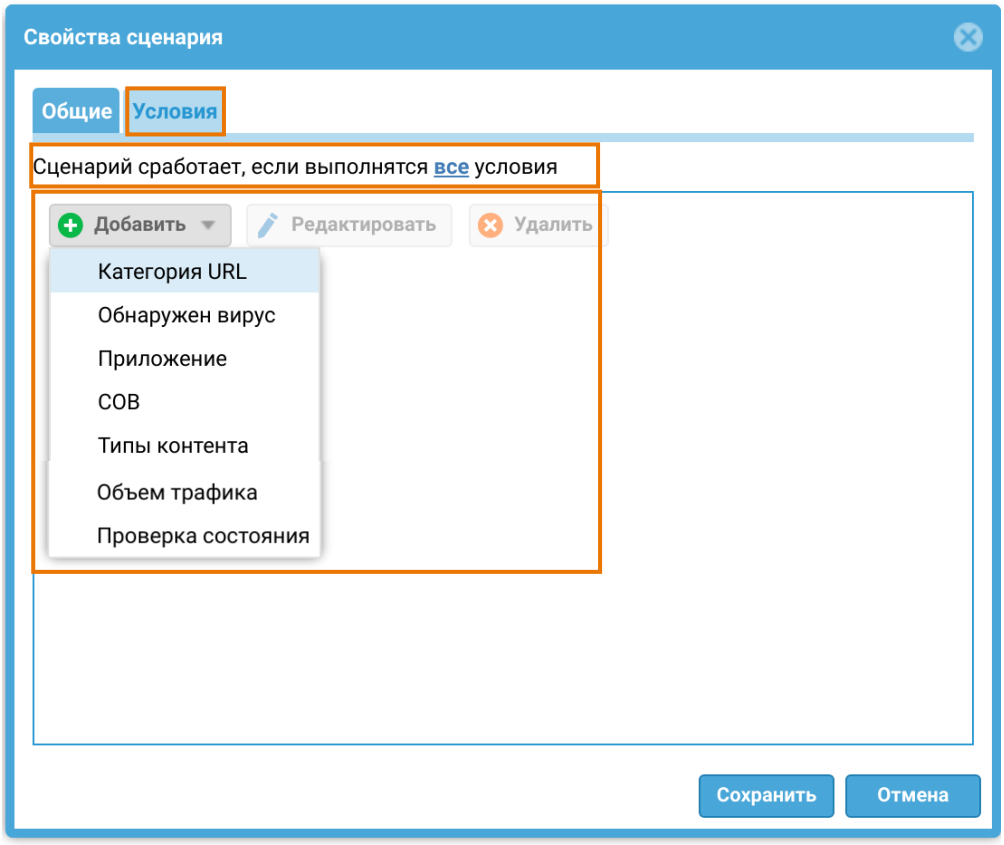
In the admin web console, scenarios are created in the **Element Libraries → Scenarios** section.

When creating a scenario, provide the following settings:



- **Enabled**: enables or disables the scenario.
- **Name**: the scenario name.
- **Description**: the scenario description.
- **Apply to**: the parameter that determines the number of users in the rule to whom the scenario will be applied. The available options are:
    - one user: the rule that uses the scenario will be applied only to the user for whom the scenario was triggered
    - all users: the rule that uses the scenario will be applied to all users listed in the rule's Users/Groups field.
- **Duration**: the duration of the restrictive rule in which the scenario was triggered.

On the **Conditions** tab, the conditions for triggering the scenario are specified. For each condition, you can specify the number of triggered events required during a certain time for the scenario to be triggered. If several conditions are set, specify whether the scenario should be triggered on matching any one of the conditions or all of them.



Setting up Scenario Trigger Conditions

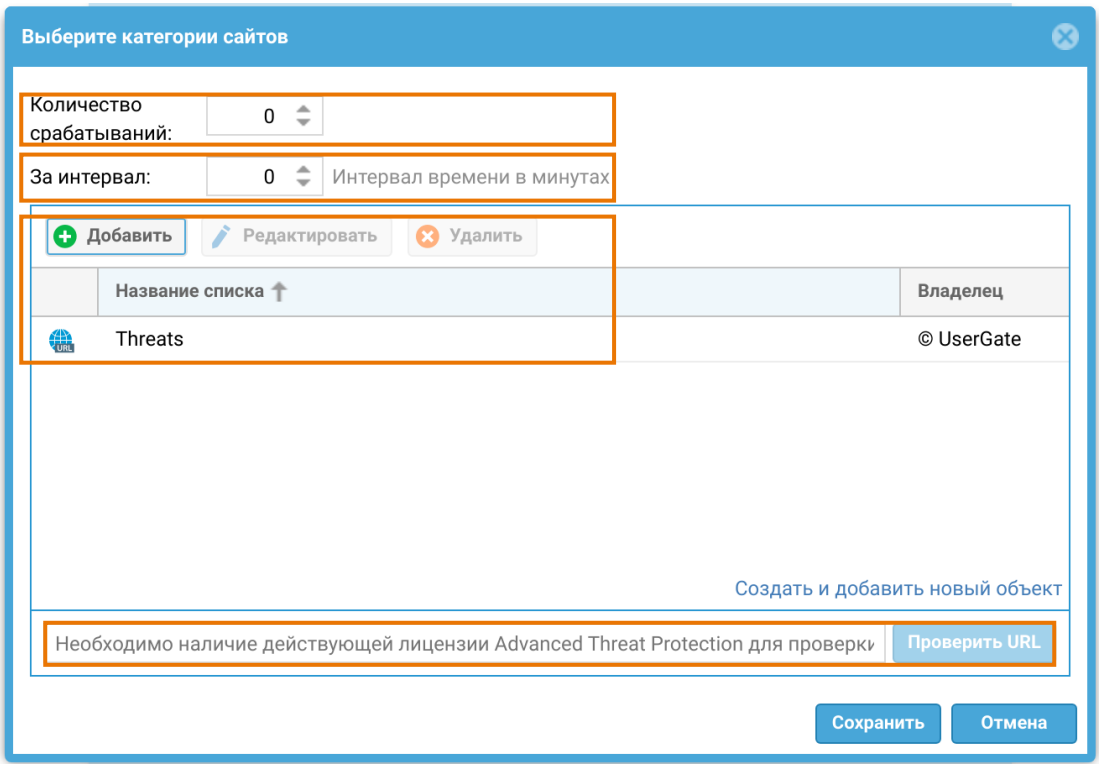
The following trigger conditions can be used in a scenario:

- **URL category**: the user's traffic matches the specified UserGate URL categories.
- **Virus detected**: the fact that a virus has been detected.
- **Application**: the specified application has been detected in the user's traffic
- **IDPS**: the intrusion prevention system has been triggered
- **Content types**: the specified content types have been detected in the user's traffic
- **Traffic limit**: the user's traffic has exceeded the limit set in the specified time frame

- **Health check**: the result of a health check for a certain resource that needs to be accessible from DCFW. (Checking can be done using the ICMP ping command, a DNS query, or an HTTP GET request).

URL category

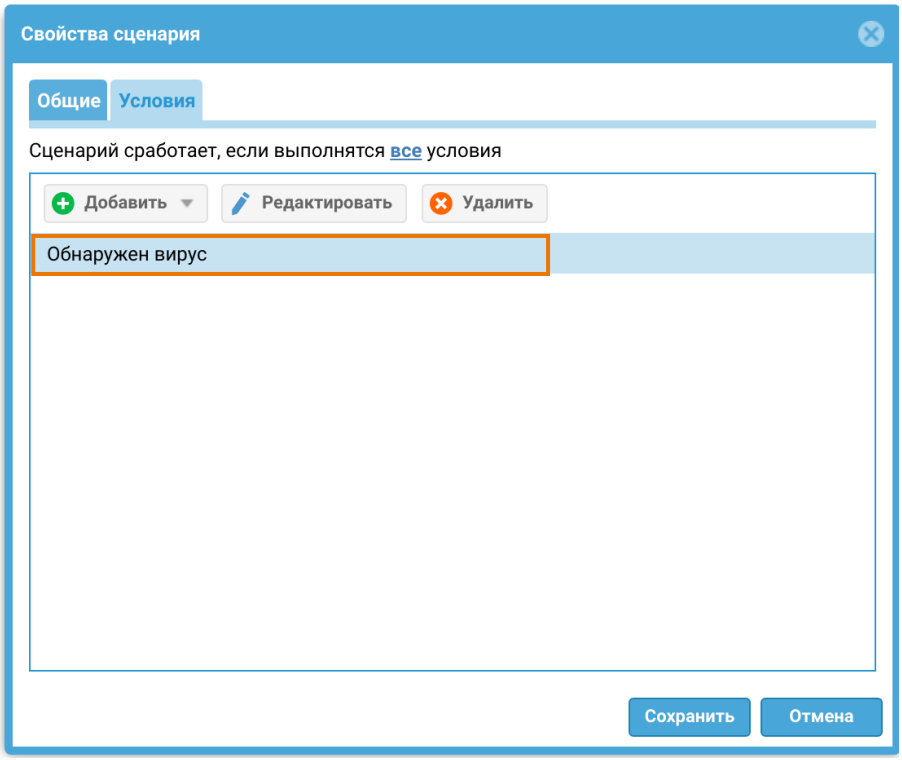
The trigger condition in this case is the match of the specified UserGate URL categories in the user's traffic.



The following parameters are configured in this condition:

- **Number of triggers**: the number of triggers after which the scenario condition is activated.
- **For interval**: the interval during which the number of triggers will be counted.
- Select website categories from the Elements library or create a list of website categories from the items in the Elements library.
- **Check URL**: the ability to check a specific URL for compliance with a particular category.

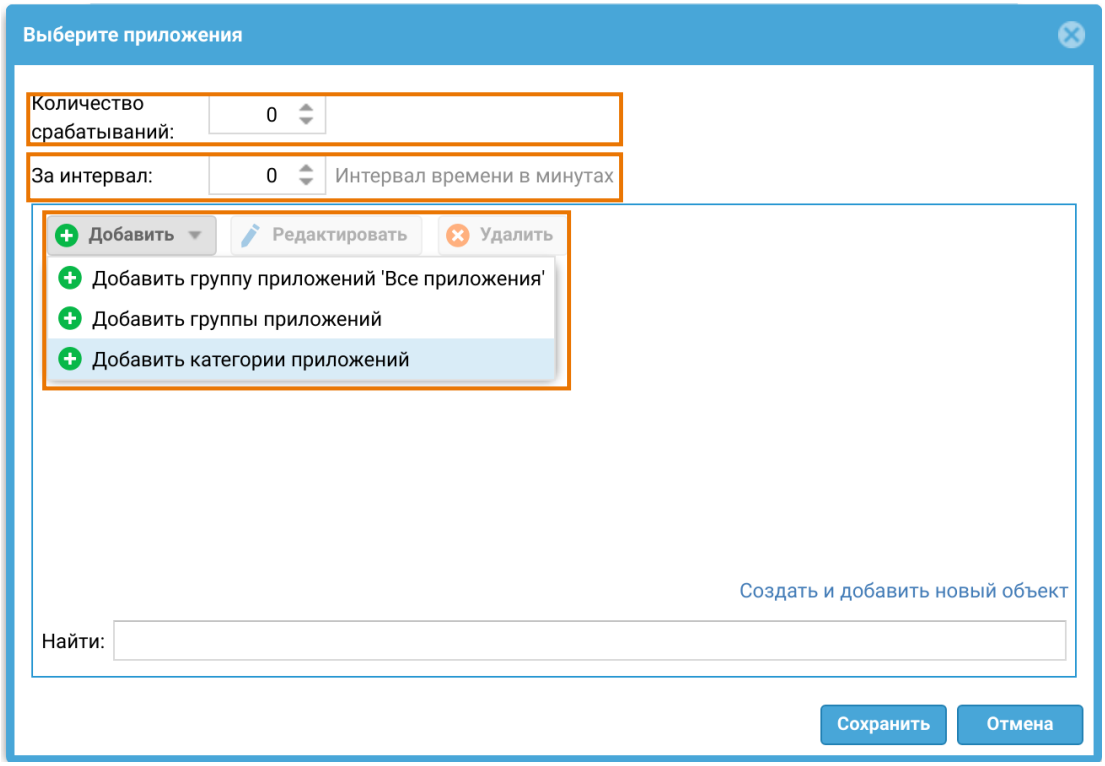
Virus Detected



The trigger condition in this case is the detection of a virus in the user's traffic.

Application

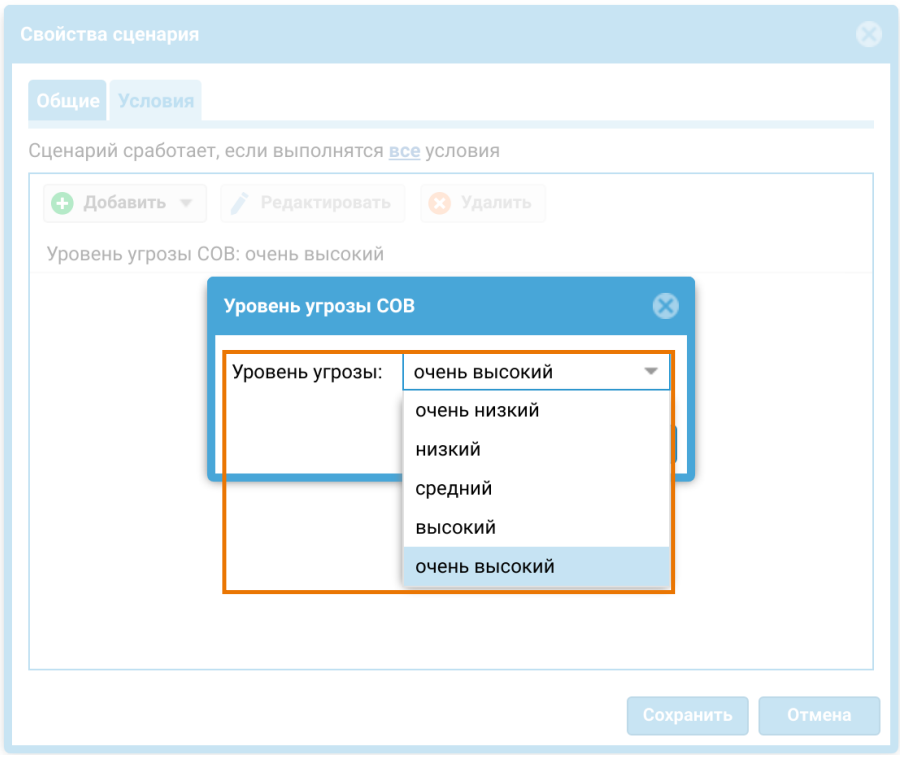
The trigger condition in this case is the detection of certain applications in the user's traffic.



The following parameters are configured in this condition:

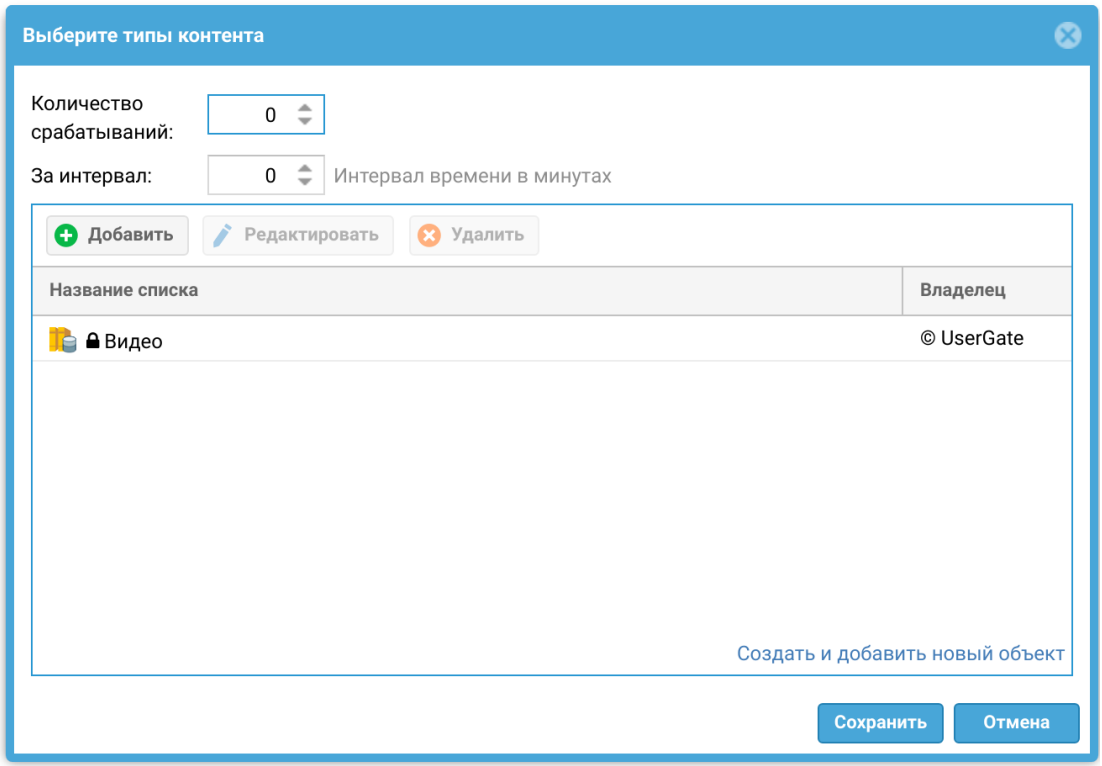
- **Number of triggers**: the number of triggers after which the scenario condition is activated.
- **For interval**: the interval during which the number of triggers will be counted.
- Select groups or categories of applications from the Elements library.

IDPS



The trigger condition in this case is the detection of a threat of a certain level by the IDPS system.

Content types

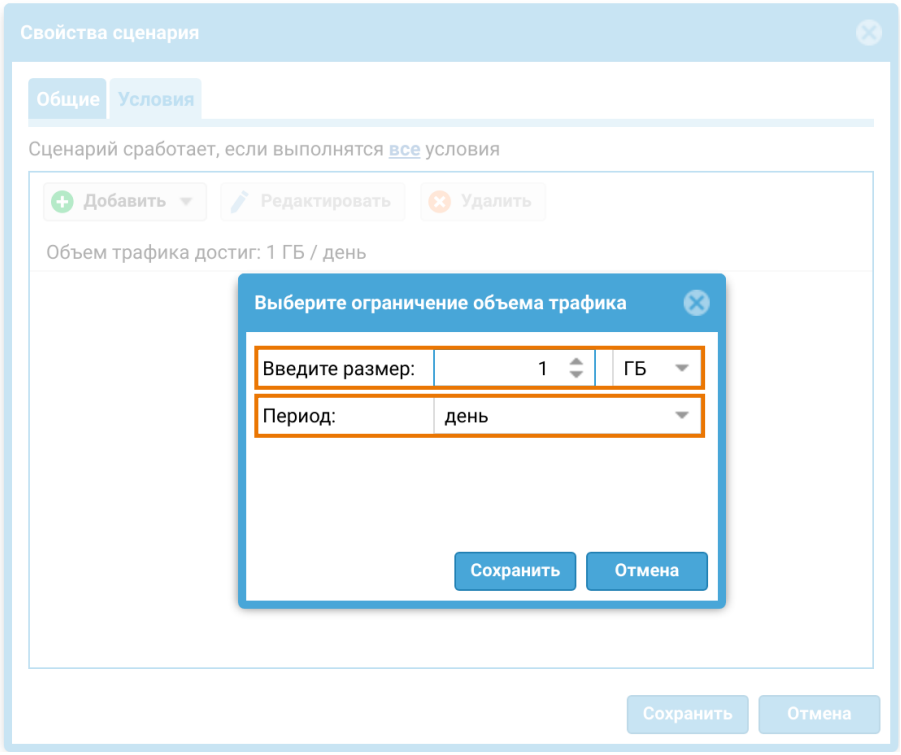


The following parameters are configured in this condition:

- **Number of triggers**: the number of triggers after which the scenario condition is activated.
- **For interval**: the interval during which the number of triggers will be counted.
- Select content types from the Elements Library.

Traffic limit

The scenario triggering condition depending on the volume of the traffic over the DCFW.



The following parameters are configured in this condition:

- **Enter size value**: the traffic limit over the DCFW which will trigger the scenario.
- **Period**: the time period for which the volume of passing traffic will be calculated.

That is, if 5GB per day is selected, then if the user's traffic exceeds 5 B in 1 day, this scenario will be triggered.

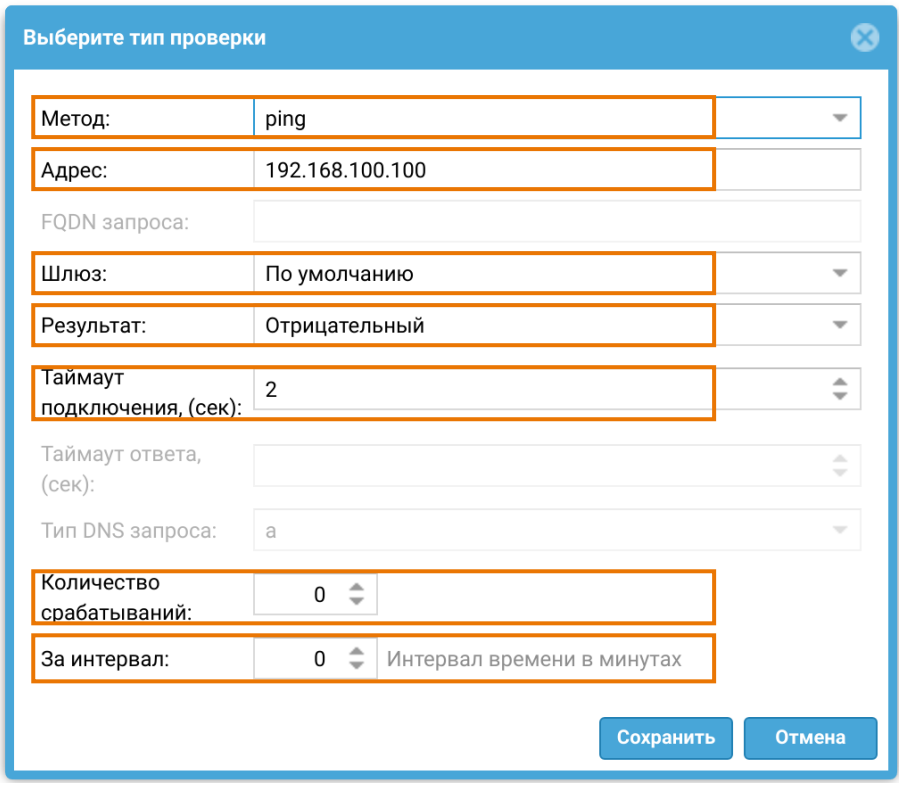
Status Check

The scenario triggering conditions depend on the server state requested by the DCFW.

The following methods are possible to check the server status:

- Ping;
- DNS;
- HTTP GET.

The **Ping** Method

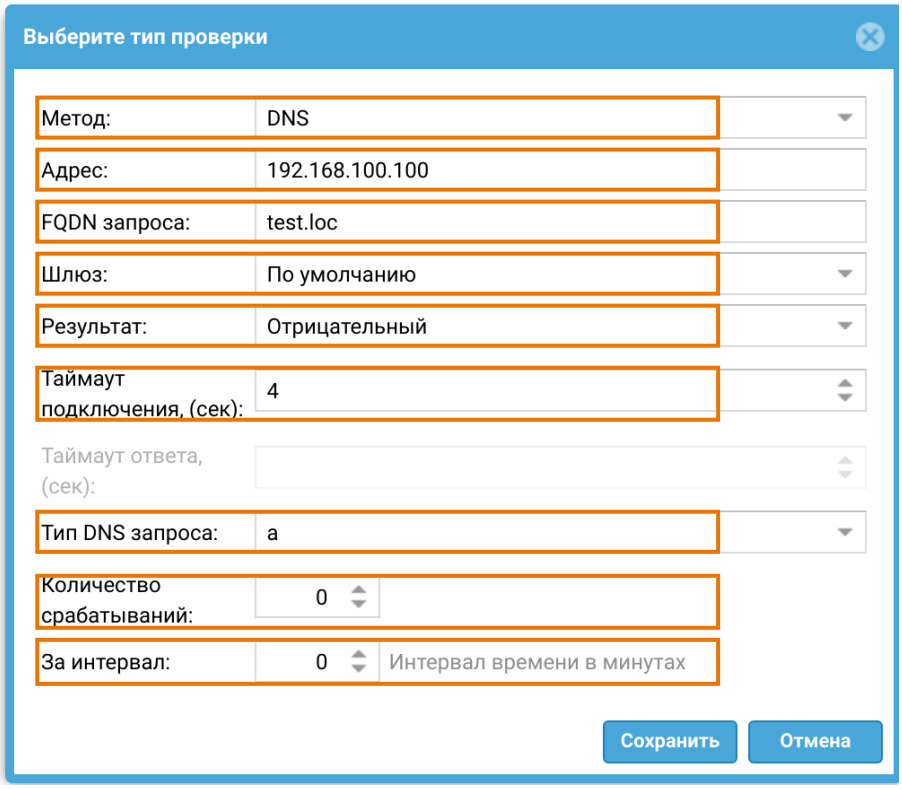


The following parameters are configured in this condition:

- **Address**: the IP address to perform ICMP ping from the DCFW.
- **Gateway**: the gateway.
- **Result**: negative or positive. Determines the expected result from pinging the server. Negative means there is no ping response; positive means there is a response.
- **Connection timeout**: the maximum time the client is willing to wait for a response from the server after a successful connection.

- **Number of triggers**: the number of triggers after which the scenario condition is activated.
- **For interval**: the interval during which the number of triggers will be counted.

The **DNS** Method

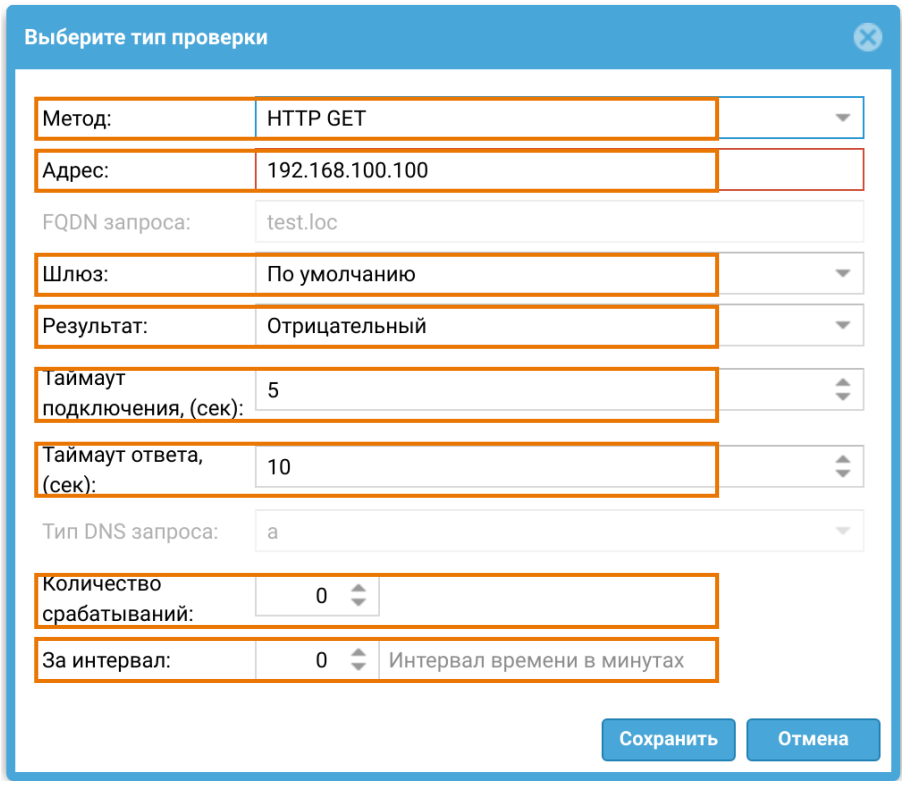


The following parameters are configured in this condition:

- **Address**: the IP address of the DNS server to which the DNS requests are sent from the DCFW.
- **Request FQDN**: the server's domain name that is resolved as part of the availability check.
- **Gateway**: the gateway.
- **Result**: positive or negative. Determines what result will be expected from the server request. Negative means there is no response; positive means there is a response.
- **Connection timeout**: the maximum time the client is willing to wait for a response from the server after a successful connection.
- **DNS request type**: the DNS request type (a, aaaa, cname, ns, ptr).

- **Number of triggers**: the number of triggers after which the scenario condition is activated.
- **For interval**: the interval during which the number of triggers will be counted.

The **HTTP GET** Method



The following parameters are configured in this condition:

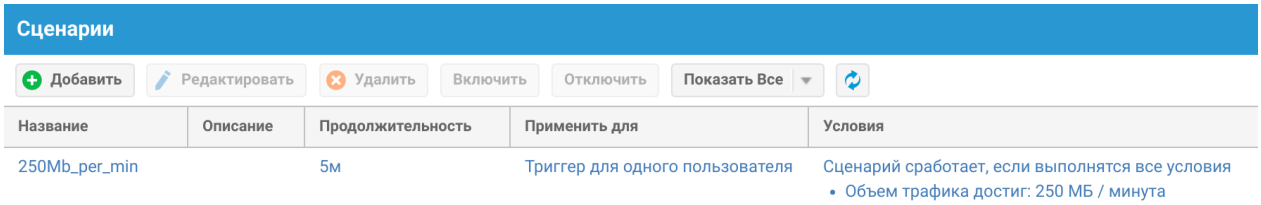
- **Address**: the domain to perform the HTTP GET request from the DCFW.
- **Gateway**: the gateway.
- **Result**: positive or negative. Determines what result will be expected from the server request. Negative means there is no response; positive means there is a response.
- **Connection timeout**: the maximum time the client is willing to wait for a response from the server after a successful connection.
- **Response Timeout**: the response timeout for checking when performing an HTTP GET.
- **Number of triggers**: the number of triggers after which the scenario condition is activated.
- **For interval**: the interval during which the number of triggers will be counted.

Example of Scenario Usage

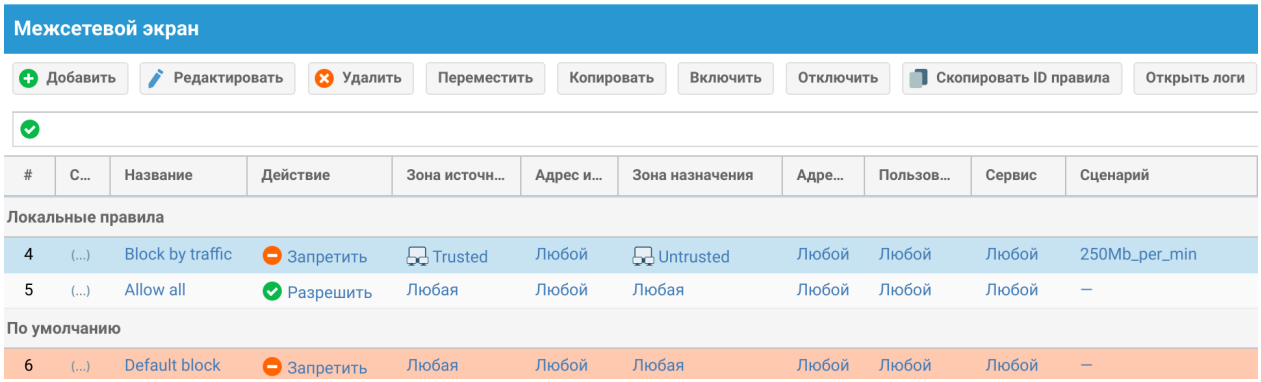
As an example, scenarios can be used in firewall rules to restrict network access if an event described in the scenario occurs.

In this example the following scenario is implemented: the firewall rule should be applied to the nodes connected to the DCFW; this rule should block access to the network for 5 minutes if the traffic volume for this note exceeds 250MB for 1 minute. Otherwise the access to the network over the DCFW should be allowed.

A scenario has been created with a trigger condition based on the volume of traffic passed:



A blocking rule has been created in the firewall, to which the created scenario has been added:



The top blocking rule **Block by traffic** in the firewall has a higher priority than the bottom allowing rule **Allow all**, but it will only work if the scenario based on the volume of traffic that has passed is triggered. In other cases, traffic will be allowed by the **Allow all** rule.

DIAGNOSTICS AND MONITORING

Traffic Monitoring

The **Traffic monitoring** section allows you to obtain a list of all user connections established through UserGate DCFW in real time. A connection is a unique combination of a source address, a destination address, and a user (if defined). For each connection, the instantaneous values of the transmit rate (TX) and receive rate (RX) are displayed. You can sort the output data by any column, and create a blocking firewall rule or a bandwidth restriction rule for the source IP address selected from the list.

> **Note**
>
> **Building this report requires a large amount of computing power on the DCFW and can result in high CPU load if a large amount of traffic is transferred. It is recommended to not keep this page open to avoid unnecessary load on the firewall.**

Routes

The **Routes** section allows you to obtain a list of all routes specified on a particular UserGate host and a particular virtual router on the cluster node. To view routes, click the **Filter** button and specify the types of route that you want to display. You can specify the following route types:

- **Connected**: routes to networks connected directly to UserGate interfaces. These routes are marked with a **C** in the route list.
- **Statically defined**: routes defined statically under **Network → Routes**. These routes are marked with an **K** in the route list.
- **OSPF**: routes received via the OSPF protocol. These routes are marked with an **O** in the route list.
- **BGP**: routes received via the BGP protocol. These routes are marked with a **B** in the route list.

The route list displayed here can be downloaded as a text file by clicking the **Export all routes** button.

OSPF

The **OSPF** section allows to get a report on the state of route channel. Using appropriate filters, you can display protocol information on a specific UserGate node and a specific virtual router on a cluster node. The following information is displayed:

- **protocol**: displays information about the parameters which are needed to configure and use OSPF on routers. (These parameters include: router ID, interface parameters, OSPF areas, routes, OSPF neighbors, information on forwarded OSPF messages and interface state, authentication secrets, time parameters and timeouts).
- **border-routers**: displays OSPF routing table records for area border router (ABR) and autonomous system border router (ASBR).
- **database**: displays information on network state and topology collected by OSPF protocol. The database stores information on routes, neighbors, interface states and other parameters.
- **route**: displays information on all routes in OSPF routing table.
- **neighbor**: displays information on a neighbor OSPF router for each interface. (This information includes neighbor ID, priority, state, TTL, idle interval, IP address, interface)

VPN

The **VPN** section displays all users and all servers connected to this server via VPN. The following information is displayed for each connection:

- **User**: username under which the connection was authenticated.
- **Server role**: client or server.
- **Session time**: duration of the established connection.
- **Tunnel IP**: IP address assigned to this client in a virtual private network.
- **IP address**: IP address from which the VPN connection was initiated.
- **Geo IP**: Geo IP country from where the connection is established.
- **Encryption**: type of encryption.

Blocked IDPS/L7 IP Addresses

The IDPS monitors and blocks attacks in real time. Preventive protection measures include connection loss, notification of the network administrator and logging to a monitoring log.

In the **Blocked IDPS/L7 IP addresses** section the list of all blocked IP addresses is displayed. Cluster nodes have the one common table **Blocked IDPS/L7 IP addresses**.

A log record includes the following parameters:

- **Blocked IP address**: contains the blocked IP address and allows to unblock and remove the IP address from the list.
- **Blocking date**: blocking time and date.
- **Signature threat**: threat level.
- **Logging status**: ability to move to logging section:
    - for trafficlog applications;
    - for idpslog IPS signature.
- **Signature details/signature name**: information on the triggered signature.
- **Destination IP**: address of the node which was attacked.
- **Blocking duration**: blocking time.
- **Time before unblocking**: remaining time countdown till the blocking is removed.

In order to unblock blocked IP addresses, select them in the list and click **Unblock**.

Packet Capture

The **Packet capture** section allows you to record the traffic that meets the specified conditions to a PCAP file for further analysis using third-party tools, such as wireshark. This may be necessary to diagnose network problems.

The section consists of three parts:

- **Filters**: here the conditions are defined under which traffic will be recorded. You can use a source address, a source port, a destination address, a destination port, an Ethernet protocol, or an IPv4 protocol as conditions. The list of IPv4 protocols can be found at the link.
- **Rules**: here UserGate interfaces are specified for traffic recording, previously created filters and the name and the size of the file where the captured traffic is recorded.
- **Files**: files with captured traffic are placed here. You can download them for analysis or delete them.

To capture traffic, perform the following steps:

Name	Description
Step 1. Create the desired filter.	Optional. You can use preinstalled filters or capture all traffic without filtering it.
Step 2. Create a rule.	Create a rule where you specify the rule name, the file name, the maximum size of the file to be written, and the necessary filters.
Step 3. Select a rule and start capturing.	Select the rule you want to use and click **Start capture**. To stop capturing, click **Stop capture**.
Step 4. Under **Files**, select a file to download.	Download the PCAP file for analysis.

Tracing Rules

Tracing rules allows administrators to see which rules are triggered when processing user HTTP(S) requests. This can be especially useful to identify problems with access to certain sites. To trace rules, do the following:

Name	Description
Step 1. Create the desired filter.	Click **Configure** in the **Diagnostics and monitoring → Tracing rule** section and specify filter parameters: • **String**: a string in user request, for example, domain name, URL, content filtering rules. • **User**: the user whose requests should be diagnosed.

Name	Description
	• **Source IP address**: IP address from which the user is making the request. The filter is necessary to limit the diagnostic information output. If it is not set, the system may also display the results of processing other users' requests.
Step 2. Start tracing.	Click **Start**
Step 3. Open the problem site.	Ask the user to open the problem site and check what rules are triggered when the site is being opened. All the rules that are executed while the user's request is being processed will be shown.

The administrator can check the content of the Internet resource displayed in the trace by using the **Open URL** form. The **Add to white list** form allows administrators to add the selected resource to one of the URL lists existing in the system.

Ping

The ping utility can be used to diagnose the availability of network resources. Ping command parameters:

Name	Description
Ping host	The host to be checked.
TTL	The maximum number of intermediate hosts allowed on the path to the host to be pinged.
Interface	The selected interface address will be used as the source address for the ping command, and the interface for sending packets will be selected in accordance with the routing table.
Counter	Number of repetitions.
Show timestamp	Add timestamps to the command output.
Don't resolve names	Use IP addresses without resolving them to domain names.

Traceroute

The traceroute utility allows you to check the path of network packets to a particular host. Traceroute parameters:

Name	Description
Traceroute host	The host to be checked.
Use ICMP	Use ICMP to execute the traceroute command. If not specified, UDP is used.
Interface	Network interface from which to execute the command.
Don't resolve names	Use IP addresses without resolving them to domain names.

DNS Query

DNS queries allow administrators to check the functioning of DNS servers.

Name	Description
DNS query (host)	DNS name to check.
Query source IP	One of the IP addresses assigned to UserGate.
DNS server	DNS server to which the query should be sent.
Port	UDP port used to make the query.
DNS query type	Type of the query.

LLDP Neighbors

This section lists all LLDP-compatible devices with enabled support for the LLDP advertisement.

Name	Description
Chassis ID	Chassis ID, a required TLV entry of the LLDP frame.

Name	Description
	Each UserGate device has only one Chassis ID. The MAC address of the management interface is used as the Chassis ID.
SysName	Name of the system.
SysDescr	Description of the system. Contains information about the hardware and the operating system of the device.
Management	Neighbor device address. Contains the following information: • IP addresses of the management interface (IPv4 and IPv6). • Interface number of the specified management address.
Capability	Device function (e.g. router, switch, etc.).
Port ID	ID of the port from which the LLDP DU (Link Layer Discovery Protocol Data Unit) was transmitted; it is a mandatory TLV entry of the LLDP frame. The MAC address of the management interface is used as the ID.
PortDescr	Description of the port.
TTL	Time to live for transmitted LLDP packets, a required TLV entry of the LLDP frame. The TTL is set in the **UserGate → Settings → Modules** section in the **Configuring LLDP** field.

LLDP Statistics

This tab displays statistics of interfaces that had LLDP profiles specified in their settings. The following information is displayed:

Name	Description
Interface	The interface name.
Transmitted	Total number of LLDP frames transmitted through the interface.
Received	Total number of LLDP frames received through the interface.
Discarded	Number of LLDP frames discarded on this interface.

Name	Description
Unrecognized	Number of LLDP frames received on this interface that contain unconfirmed content.
Ageout	Each LLDP frame contains information on how long the LLDP information is valid (the ageout). If no new frames are accepted during the ageout period, the LLDP information is deleted.
Inserted	Number of added records containing information on LLDP neighbors.
Deleted	Number of deleted records containing information on LLDP neighbors.

NOTIFICATIONS

SNMP

UserGate supports monitoring using the SNMP v2c and SNMP v3 protocols. Both SNMP queries and SNMP trap management are supported. This allows you to monitor critical UserGate parameters using the SMNP management software used in your company.

To configure monitoring using SNMP:

1. In the properties of the zone of the interface to which the connection will be made via the SNMP protocol, in the **Access control** tab, enable the **SNMP** service.

2. Create an SNMP rule.

To create an SNMP rule, click the **Add** button under **SNMP** and specify the following parameters:

Name	Description
Rule name	The name of the rule.
Server IP address for traps	The IP address of the trap server and the port on which the server will listen for notifications. Usually, it is UDP port 162.

Name	Description
	This setting is required only if you need to send traps to the notification server.
Community	SNMP community — the string for UserGate server identification and SNMP server identification for SNMP v2c. Use only Latin letters and numbers.
Context	Optional parameter that defines the SNMP context. Use only Latin letters and numbers. Some devices may have multiple copies of the entire MIB subtree. For example, several virtual routers can be created on the device. Each such virtual router will have a complete MIB subtree. In this case, each virtual router can be specified as a context on the SNMP server. The context is identified by name. When the client makes a request, the context name can be specified. If the context name is not specified, the default context will be requested.
Version	Specify the version of the SNMP protocol used in the rule. Available options: SNMP v2c and SNMP v3.
Allow SNMP queries	When enabled, allows receiving and processing of SNMP requests from the SNMP manager.
Allow SNMP traps	When enabled, allows sending of SNMP traps to the server configured to receive notifications.
SNMP security profile name	For SNMP v3 only. For more details, see the SNMP Security Profiles section.
Events	Selecting the types of parameters available for monitoring by rule.

Note

Authentication settings for SNMP v2c (community) and SNMP v3 (user, authentication type, authentication algorithm, authentication password, encryption algorithm, encryption password in SNMP security profile) on the SNMP manager must match those of UserGate.

For information on configuring authentication settings for your SNMP manager, refer to the configuration guide for your SNMP management software.

UserGate is assigned the unique **SNMP PEN** (Private Enterprise Number) **45741**.

You can download current UserGate MIB files with monitoring parameters from the device administrator console. To do this, go to the **Diagnostics and monitoring** tab, then click **Download MIB** in the **Notifications → SNMP** section

You can download the following MIB files:

- UTM-TRAPS-MIB
- UTM-TRAPS-BINDINGS-MIB
- UTM-MIB
- UTM-INTERFACES-MIB.
- UTM-TEMPERATURE-MIB.

UTM-TRAPS-MIB

Name	Description
trapCoreCrush	Core crash.
trapStatDown	Statistics service (UserGate Log Analyzer) unavailable.
trapCoreBootstrapEnd	Server booting has finished successfully.
trapDefaultGatewayChanged	Default gateway has been changed.
trapHighSessionsCounter	Conntrack table 90% full.
trapHighUsersCounter	Number of active users has reached 90% of the license threshold.
trapDataPartitionFSStatus	File system status. The file system status changed to "not_clean".
trapStatusChanged	Status of the HA cluster node has been changed.
trapMemberUp	Status of the HA cluster node has been changed to "Connected".
trapMemberDown	HA cluster node has been disconnected.
trapAttackDetected	Detection of an attack by the IDPS.
trapChecksumFailed	Binary files checksum mismatch.

Name	Description
trapHighCPUUsage	High CPU usage (95%).
trapLowMemory	High memory usage (95%).
trapLowLogdiskSpace	Not enough disk space to store logs.
trapRaidStatus	RAID status has been changed.
trapPowerSupply	The first power supply is off.
trapCableStatus	Cable has been connected or disconnected from the interface.
trapHighDiskIOUtilization	High disk load. An alert is sent when the load is >=95% in 5 minutes on at least one of the disk devices.
trapTrafficDrop	A firewall deny rule has been triggered.
trapLDAPServerDown	LDAP server unavailable.
trapCriticalTemperature	Critical temperature on one of the sensors. An alert is sent when one of the operating temperature limits (lower or upper) is crossed. The lower limit of operating temperature is usually 0°C (-40°C for X series devices), the upper limit is 85°C.

UTM-TRAPS-BINDINGS-MIB

Name	Data type	Description
utmSessions	Integer	Current number of active sessions.
utmSessionsMax	Integer	Maximum number of active sessions.
utmUsers	Integer	Current number of active users.
utmUsersMax	Integer	Maximum number of active users.
utmDataPartionFSStatus	Integer	File system status. • **0** — clean. • **1** — not clean.
utmHAStatus	Integer	

Name	Data type	Description
		Current status of the HA cluster node: • **0**: master node • **1**: slave node • **3** — fault.
utmHAStatusReason	Integer	Reason for the change of the HA cluster node status: • **1**: connection to the node has been lost • **2**: HTTP proxy server unreachable • **3**: no reachable gateway • **4**: DNS server unreachable • **5**: UserGate Management Center node is unreachable.
utmCPUUsage	Integer	CPU load (in %).
utmMemory	Integer	RAM usage (in %).
utmLogdiskSpace	Integer	Disk space used for logs (in %).
utmAdaptecRaidStatus	Integer	Current status of RAID (Redundant Array of Independent Disks) built on the Adaptec controller: • **no_raid**. • **0**: optimal: the array is in its optimal state • **1**: degraded: one drive has completely or partially failed. • **2**: rebuild: array rebuild in progress
utmBroadcomRaidStatus	Integer	Current status of RAID (Redundant Array of

Name	Data type	Description
		Independent Disks) built on the Broadcom controller: • **no_raid** • **0**: optimal: the array is in its optimal state • **1**: degraded: one drive has completely or partially failed. This status occurs if 2 disks fail. • **2**: partialDegraded: one drive has completely or partially failed. • **3**: failed: not operable due to an error • **4**: offline: drive is not available to the RAID controller
utmPowerSupply	Integer	Number of power supplies: • **1**: one power supply • **2**: two power supplies
utmPowerSupplyStatus	Integer	State of the power supply: • **no_power_supplies**. • **0** — off. • **1** — on.
utmCSCIfName	String	The interface name.
utmCSCStatus	Integer	Status of the network adapter: • **1**: cable connected • **2**: cable disconnected
utmDiskIOUtilization	Integer	Current disk utilization (%).
utmLDAPServerName	String	LDAP server name.
utmLDAPServerAddress	String	LDAP server IP address.

Name	Data type	Description
utmThermSensor	String	Name of the temperature sensor.
utmThermValue	Integer	Temperature value measured by the sensor.

UTM-MIB

Name	Data type	Description
vcpuCount	Integer	Number of virtual CPUs in the system.
vcpuUsage	Integer	System virtual processor load; displays the actual number of virtual processors loaded.
usersCounter	Integer	Current number of active users. (*)
sessionsCounter	Integer	Current number of active sessions. (*)
tcpsessionsCounter	Integer	Current number of active TCP sessions. (*)
udpsessionsCounter	Integer	Current number of active UPD sessions. (*)
icmpsessionsCounter	Integer	Current number of active ICMP sessions. (*)
sessionsRate10	Integer	Number of new sessions per second. Average value for the last 10 seconds. (*)
sessionsRate60	Integer	Number of new sessions per second. Average value for the last 60 seconds. (*)
sessionsRate300	Integer	Number of new sessions per second. Average value for the last 300 seconds. (*)
tcpsessionsRate10	Integer	Number of new TCP sessions per second. Average value for the last 10 seconds. (*)

Name	Data type	Description
tcpsessionsRate60	Integer	Number of new TCP sessions per second. Average value for the last 60 seconds. (*)
tcpsessionsRate300	Integer	Number of new TCP sessions per second. Average value for the last 300 seconds. (*)
udpsessionsRate10	Integer	Number of new UPD sessions per second. Average value for the last 10 seconds. (*)
udpsessionsRate60	Integer	Number of new UPD sessions per second. Average value for the last 60 seconds. (*)
udpsessionsRate300	Integer	Number of new UPD sessions per second. Average value for the last 300 seconds. (*)
icmpsessionsRate10	Integer	Number of new ICMP sessions per second. Average value for the last 10 seconds. (*)
icmpsessionsRate60	Integer	Number of new ICMP sessions per second. Average value for the last 60 seconds. (*)
icmpsessionsRate300	Integer	Number of new ICMP sessions per second. Average value for the last 300 seconds. (*)
dnsRequestCounter	Integer	Total DNS requests. (*)
dnsBlockedRequestCounter	Integer	Blocked DNS requests. (*)
dnsRequestRate	Integer	DNS requests per second. (*)
httpRequestCounter	Integer	Total HTTP requests. (*)
httpBlockedRequestCounter	Integer	Blocked HTTP requests. (*)
httpRequestRate	Integer	HTTP queries per second. (*)
dataPartitionFSStatus	String	File system status.

Name	Data type	Description
haStatus	Integer	The current state of the cluster node.
cpuLoad	Integer	System CPU load (in %).
memoryUsed	Integer	RAM usage (in %).
logDiskSpace	Integer	Disk space used for logs (in %).
powerSupply1Status	String	State of the first power supply: • **no_power_supplies**. • **on** • **off**
powerSupply2Status	String	State of the second power supply: • **no_power_supplies**. • **on** • **off**
raidType	String	RAID array type.
raidStatus	String	Current status of RAID (Redundant Array of Independent Disks): • **no_raid**. • **0**: optimal: the array is in its optimal state • **1**: degraded: one drive has completely or partially failed. • **2**: rebuild: array rebuild in progress
diskIOUtilization	Integer	Current disk utilization (%).
diskIOUtilization60	Integer	Disk utilization (%). Average value for the last 60 seconds.
diskIOUtilization300	Integer	

Name	Data type	Description
		Disk utilization (%). Average value for the last 300 seconds.

> **Note**
> **Metrics marked with the (*) symbol in the description are not relevant for *UGMC* and *LogAn*. Metric values for these devices will always be zero.**

UTM-INTERFACES-MIB

Name	Data type	Description
ifNumber	Integer	Number of network interfaces.
ifIndex	Integer	The value is unique for each interface. Available values: from 1 to ifNumber.
ifDescr	String	Interface description.
ifType	Integer	Interface type determined according to the physical/link layer protocol: • **1**: other: unknown type • **2**: regular1822: defined in BBN Report 1822 • **3**: hdh1822: defined in BBN Report 1822 • **4**: ddn-x25: defined in BBN Report 1822 • **5**: defined in the data link layer standard of the OSI X.25 network model • **6**: ethernet-csmacd: Ethernet-type network interface regardless of speed (defined in RFC 3635) • **7**: iso88023-csmacd: defined in IEEE 802.3

Name	Data type	Description
		• **8**: iso88024-tokenBus: defined in IEEE 8802.4 • **9**: iso88025-tokenRing: network interface uses a Token Ring connection; defined in the IEEE 802.5 standard. • **10**: iso88026-man: defined in the ISO 88026 standard "MAN". • **11**: starLan: defined in the IEEE 802.3e standard. • **12** — proteon-10Mbit — Proteon 10 Mbit. • **13** — proteon-80Mbit — Proteon 80 Mbit. • **14**: hyperchannel: high-speed channel used in ISDN networks. • **15**: fddi: network interface uses FDDI (Fiber Distributed Data Interface) connection. FDDI is a set of standards for data transmission over fiber-optic lines in local networks. • **16**: lapb: data link layer protocol used to transmit X.25 standard packets. • **17**: sdlc: data link layer protocol for IBM system network architecture. • **18**: ds1: can handle 24 simultaneous connections at a total speed of 1.544Mbit/s; also called T1. • **19**: e1: European equivalent of T1. • **20**: basicISDN: used for communication between the

Name	Data type	Description
		subscriber's equipment and the ISDN station. • **21**: primaryISDN: used to connect to broadband backbones, connecting local and central PBX or network switches. • **22**: propPointToPointSerial: defined in RFC1213. • **23**: ppp: network interface uses PPP (Point-To-Point Protocol) connection. • **24**: softwareLoopback: network interface configured as a loopback adapter. These interfaces are often used for testing; they do not send traffic to the network. • **25**: eon: ConnectionLess Network Protocol (CLNP) over Internet Protocol (IP); defined in ISO/IEC 8473-1. • **26**: ethernet-3Mbit: network interface uses a 3Mbit/s Ethernet connection. This version of Ethernet is defined in the IETF standard RFC 895. • **27**: nsip, XNS over IP: intended for use in a variety of data transmission environments. • **28**: slip: network interface uses a SLIP (Serial Line Internet Protocol) connection. SLIP is defined in the IETF RFC 1055 standard.

Name	Data type	Description
		• **29** — ultra — ULTRA Technologies. • **30**: ds3: high-speed data interface multiplexing DS1 and DS2 signals; also know as T3. • **31**: sip: network interface uses a SLIP (Serial Line Internet Protocol) connection. SLIP is defined in the IETF RFC 1055 standard. • **32**: frame-relay: allows packet-switched data transmission across an interface between user devices and network equipment.
ifMtu	Integer	Maximum size of a network layer packet that can be sent over this interface.
ifSpeed	gauge32	Interface bandwidth in bits per second.
ifPhysAddress	String	Physical interface address (MAC address).
ifAdminStatus	Integer	Interface state assigned by the administrator: • **1**: up: ready to transmit packets • **2**: down: not working • **3**: testing: working in the test mode; cannot transmit work packets.
ifOperStatus	Integer	Current operating status of the interface: • **1**: up: ready to transmit packets

Name	Data type	Description
		• **2**: down: interface cannot transmit data packets • **3**: testing: network interface is being tested; cannot transmit working packets • **4**: unknown: interface state is unknown • **5**: dormant: network interface cannot transmit data packets, it is waiting for an external event • **6**: notPresente: network interface cannot transmit data packets because a component, usually a piece of hardware, is missing • **7**: lowerLayerDown: network interface cannot transmit data packets because it is running on top of one or more other interfaces, and at least one of those "lower-layer" interfaces is down
ifLastChange	timeticks	SysUpTime value when the interface switches to this state.
ifInOctets	counter32	Number of bytes received by the interface, including service bytes.
ifInUcastPkts	counter32	Number of delivered unicast packets.
fInNUcastPkts	counter32	Number of delivered multicast and broadcast packets.
ifInDiscards	counter32	Number of incoming packets that were dropped, even if no

Name	Data type	Description
		errors were detected preventing the delivery. Buffer space release may be one of the reasons for dropping.
ifInErrors	counter32	Number of incoming packets that contain errors preventing the delivery.
ifInUnknownProtos	counter32	Number of packets that were received through the interface and dropped because an unknown or unsupported protocol was used.
ifOutOctets	counter32	The number of bytes transmitted by the interface, including service bytes.
ifOutUcastPkts	counter32	Number of sent unicast packets, including packets that were dropped or not sent.
ifOutNUcastPkts	counter32	The number of sent multicast and broadcast packets, including packets that were dropped or not sent.
ifOutDiscards	counter32	Number of outgoing packets that were dropped, even if no errors were detected preventing the transmission. Buffer space release may be one of the reasons for dropping.
ifOutErrors	counter32	The number of outgoing packets that could not be transmitted due to errors.
ifOutQLen	gauge32	The send queue length (number of packets).
ifInMulticastPkts	counter32	Number of delivered multicast packets.
ifInBroadcastPkts	counter32	Number of delivered broadcast packets.

Name	Data type	Description
ifOutMulticastPkts	counter32	Number of sent multicast packets, including packets that were dropped or not sent.
ifOutBroadcastPkts	counter32	Number of sent broadcast packets, including packets that were dropped or not sent.
ifHCInOctets	counter64	Identical to **ifInOctets**: number of bytes received by the interface, including service bytes; uses a higher capacity counter.
ifHCInUcastPkts	counter64	Identical to **ifInUcastPkts**: number of delivered unicast packets; uses a higher capacity counter.
ifHCInMulticastPkts	counter64	Identical to **ifInMulticastPkts**: number of delivered multicast packets; uses a higher capacity counter.
ifHCInBroadcastPkts	counter64	Identical to **ifInBroadcastPkts**: number of delivered broadcast packets; uses a higher capacity counter.
ifHCOutOctets	counter64	Identical to **ifOutOctets**: number of bytes transmitted by the interface, including service bytes; uses a higher capacity counter.
ifHCOutUcastPkts	counter64	Identical to **ifOutUcastPkts**: number of sent unicast packets, including packets that were dropped or not sent; uses a higher capacity counter.
ifHCOutMulticastPkts	counter64	Identical to **ifOutMulticastPkts**: number of sent multicast packets, including packets that were dropped or not

Name	Data type	Description
		sent; uses a higher capacity counter.
ifHCOutBroadcastPkts	counter64	Identical to **ifOutBroadcastPkts**: number of sent broadcast packets, including packets that were dropped or not sent; uses a higher capacity counter.
ifLinkUpDownTrapEnable	Integer	Specifies whether to create a trap when the link status changes: • **1**: enabled • **2**: disabled
ifHighSpeed	gauge32	Current estimated interface bandwidth pool in bit/s, kbit/s, Mbit/s, or Gbit/s.
ifPromiscuousMode	Integer	Promiscuous mode. Available values: • **1**: true: station receives all packets/frames regardless of the destination. • **2**: false: interface receives only packets/frames addressed to this station. The object value does not affect the reception of broadcast and multicast packets/frames.
ifAlias	String	Interface name assigned by the administrator.
ifCounterDiscontinuityTime	timeticks	SysUpTime value when the event occurred that caused one or more interface counters to fail.

UTM-TEMPERATURE-MIB

Name	Data type	Description
termNumber	Integer	Number of temperature sensors on this platform.
thermLowerThreshold	Integer	Lower operating temperature limit.
thermUpperThreshold	Integer	Upper operating temperature limit.
thermTable	sequence	Table of temperature sensors with readings (thermEntry).
thermEntry	sequence	A specific sensor info: • thermName (string): sensor name. • thermValue (integer): sensor readings. • thermUnit (string): sensor reading unit.

Note

Temperature sensor data will only be displayed for supported hardware platforms. Currently supported devices are UserGate C150, C151, FG, X10. For unsupported platforms or virtual solutions, the sensor table will be empty, and the number of sensors and operating temperature limits will be zero.

Note

If taking a temperature reading from a sensor was not possible, it will not be transmitted in the table, while the thermNumber parameter counts the total number of temperature sensors, even taking into account those that are not working. In this case, the number of sensors in the table and the thermNumber value may not match.

SNMP Parameters

This section allows to specify parameters of providing information over SNMP protocol by the SNMP agent. SNMP parameters are specified for each node separately.

Name	Description
SNMP system name	Name of the system which is used by SNMP control subsystem.
SNMP system location	Information on physical location of the SNMP agent.
SNMP system description	Description of the system.
Engine ID	Each UserGate device has a unique SNMPv3 Engine ID. By default, the Engine ID is generated from the UserGate node name. When editing the Engine ID, you are required to specify its length, type, and value. The length can be defined as **fixed** (max. 8 bytes) or **dynamic** (max. 27 bytes). A fixed ID length is only applicable to the **text** type. The Engine ID can be generated in these formats: • IPv4 (ip4) • IPv6 (ipv6) • MAC address (mac) • Text (text) • Octets (octets).

SNMP Security Profiles

In this section the security profiles for the SNMPv3 manager authentication are configured.

Note

SNMP v3 authentication parameters (username, password, authentication type and algorithm, encryption algorithm and password) at the SNMP manager should match SNMP parameters in UserGate.

Name	Description
Name	SNMP security profile name
Description	SNMP security profile description
User	User name to authenticate the SNMP manager.
Authentication type	Select an authentication mode for the SNMP manager. The available options are: • No authentication; No encryption (noAuthNoPriv) • Authentication; No encryption (authNoPriv) • Authentication; Encryption (authPriv). The authPriv mode is considered the most secure.
Authentication algorithm	The algorithm used for authentication. Possible to use: • SHA1 • MD5 • SHA224 • SHA256 • SHA384 • SHA512
Authentication password	The password used for authentication.
Encryption algorithm	The algorithm used for encryption. DES or AES can be used.
Encryption password	The password used for encryption.

Alert Rules

This section allows you to define alert rules, which can be used to send notifications about different types of events, for example, a high CPU load or a password sent to the user by SMS. To create an alert rule, follow these steps:

Name	Description
Step 1. Create one or more notification profiles.	See the Notification Profiles section.
	See the Emails and Phones sections.

Name	Description
Step 2. Create alert recipient groups.	
Step 3. Create an alert rule.	Add a rule on the **Diagnostics and monitoring** tab in the **Notifications → Alert rules** section.

Specify the following parameters for the rule:

Name	Description
Enabled	Enables or disables the rule.
Name	The name of the rule.
Description	A description of the rule.
Notification profile	A previously created notification profile. For SMPP profiles, a tab will open where you can specify recipients as phone numbers. For SMTP profiles, a tab will open where you can specify recipients as email addresses.
From	From whom the notifications will come.
Subject	Notification subject.
Timeout before resending (in seconds)	Specify the timeout during which the server will not send a message when this rule is triggered again. This setting prevents a flood of messages when an alert rule is triggered frequently.
Events	Specify events for which you want to receive alerts.
Phones	For SMPP profiles, specify the phone groups to which SMS notifications will be sent.
Emails	For SMTP profiles. specify groups of email addresses to which email notifications will be sent.

LOGS AND REPORTS

LOGS

General Information

UserGate DCFW logs all events that occur during its operation, and registers them in the following logs:

- **Events**: events related to changes in DCFW settings, user and administrator authentication, updates to various lists, etc.
- **Web access**: a detailed log of all web requests processed by DCFW.
- **DNS**: contains events related to the DNS traffic.
- **Traffic**: detailed log of all firewall, NAT, DNAT, Port forwarding, and Policy-based routing rules triggered. To log these events you need to enable logging in the required rules for the firewall, NAT, DNAT, Port forwarding, or Policy based routing.
- **IDPS**: events logged by the intrusion detection and prevention system.
- **Search history**: user search queries in popular search engines.
- **UserID agent**: contains description of events showing the result of the UserID agent's work.

Log management is automated: logs are cyclically overwritten, providing free disk space necessary for work.

Log records (except the event log) are rotated automatically based on the free space on a given partition. Database rotation records will be displayed in the event log. If LogAn is connected, then the record will be displayed in the LogAn event log.

Event log records are never rotated.

Event Log

The event log displays events related to changes to the DCFW settings, such as added / deleted / edited account data, rules, or other items. It also displays all

events of web console login, user authentication through the Captive portal or VPN, server start, shutdown, restart, etc.

To assist in finding the events of interest, the records can be filtered by various criteria such as the date range, component, severity, or event type.

Administrators can select to display only the columns they need. To do this, click on any of the columns and set the checkmarks for the columns you want to display in the context menu that appears.

By clicking **Export as CSV**, the administrator can save the filtered log data in a .csv file for subsequent analysis.

Click **Show** to open a window with a detailed event description.

Web Access Log

The Web access log displays all user requests to the Internet via HTTP and HTTPS. It displays events that triggered content filtering, SSL inspection, web security, and Captive portal rules that have logging enabled. The following information is displayed:

- DCFW node where the event occurred.
- Event time
- Event details
- User
- Action
- Rule
- Reasons (if a site is blocked)
- Destination URL
- Source zone
- Source IP address
- Source port
- Destination zone

- Destination IP address
- Destination port
- URL categories.
- Application
- Application layer protocol
- HTTP method
- Status code.
- Content type (if present)
- Information
- Bytes sent/received
- Packets sent/received
- Referrer (if present)
- Operating system
- Useragent.

Administrators can select to display only the columns they need. To do this, click on any of the columns and set the checkmarks for the columns you want to display in the context menu that appears.

To assist in finding the events of interest, the records can be filtered by various criteria such as the user account, rule, action, etc.

By clicking **Export as CSV**, the administrator can save the filtered log data in a .csv file for subsequent analysis.

Click **Show** to open a window with a detailed event description.

DNS Log

DNS log lists events related to the DNS traffic. To log DNS events on the DCFW, DNS filtering must be enabled in the DNS proxy settings and logging must be enabled in the content filtering rules where DNS traffic is logged.

The following information is displayed:

- Node
- Time
- User
- Rule
- Reasons
- Domain name
- Source zone
- Source IP address
- Source port
- Source MAC address.
- Destination zone
- Destination IP address
- Destination port
- Network protocol
- URL category.
- Information

Administrators can select to display only the columns they need. To do this, click on any of the columns and set the checkmarks for the columns you want to display in the context menu that appears.

To assist in finding the events of interest, the records can be filtered by various criteria such as the protocol, date range, action, etc.

By clicking **Export as CSV**, the administrator can save the filtered log data in a .csv file for subsequent analysis.

Click **Show** to open a window with a detailed event description.

Traffic Log

The Traffic log displays firewall and NAT rule trigger events for rules where logging is enabled. The following information is displayed:

- DCFW node where the event occurred.
- Event time
- Event details
- User
- Action
- Rule
- Application
- Network protocol
- Source zone
- Source IP address
- Source port
- Source MAC address
- Destination zone
- Destination IP address
- Destination port
- Destination MAC address
- NAT source IP address (in case of a NAT rule)
- NAT source port (in case of a NAT rule)
- NAT destination IP address (in case of a NAT rule)
- NAT destination port (in case of a NAT rule)
- Bytes sent/received
- Packets sent/received

Administrators can select to display only the columns they need. To do this, click on any of the columns and set the checkmarks for the columns you want to display in the context menu that appears.

To assist in finding the events of interest, the records can be filtered by various criteria such as the user account, rule, action, etc.

By clicking **Export as CSV**, the administrator can save the filtered log data in a .csv file for subsequent analysis.

Click **Show** to open a window with a detailed event description.

IDPS Log

The intrusion detection system log displays the triggered IPS signatures for which the logging or blocking action has been set. The following information is displayed:

- PCAP files
- DCFW node where the event occurred.
- Time
- Event details
- User
- Action
- Rule
- Signatures
- Application
- Network protocol
- Source zone
- Source IP address
- Source port
- Source MAC address
- Destination zone

- Destination IP address
- Destination port
- Destination MAC address

Administrators can select to display only the columns they need. To do this, click on any of the columns and set the checkmarks for the columns you want to display in the context menu that appears.

To assist in finding the events of interest, the records can be filtered by various criteria such as the protocol, date range, action, etc.

By clicking **Export as CSV**, the administrator can save the filtered log data in a .csv file for subsequent analysis.

Click **Show** to open a window with a detailed event description.

Search History

The **Search history** section displays all user search queries that are configured to be logged in the safe browsing policies. Administrators can select to display only the columns they need. To do this, click on any of the columns and set the checkmarks for the columns you want to display in the context menu that appears.

To assist in finding the events of interest, the records can be filtered by various criteria such as users, date range, search engines, etc.

By clicking **Export as CSV**, the administrator can save the filtered log data in a .csv file for subsequent analysis.

Click **Show** to open a window with a detailed event description.

UserID Agent

Windows Active Directory log

The Windows Active Directory log displays events collected by the UserID agent using the "Microsoft Active Directory" connector from AD servers. The log displays successful login events (event ID 4624), Kerberos events (event IDs 4768, 4769, 4770),

and group membership events (event ID 4627). The log contains the following information:

Name	Description
Node	UserGate node where the event occurred.
Time	The time of the event.
Endpoint event log record details	The link to the event.
Device/sensor	UserID connector.
Log level	The "Keywords" field from the AD log.
Data	Event details from AD log.
Log event source	The "Source" field from the AD log.
Log category	Incident category code (12554 Group Membership, 12544 Logon, 14337 Kerberos Service Ticket Operations etc.)
Incident category	The "Task type" field from the AD log.
Computer name	windows node where the event took place.
User	The "User" field from AD log.
Log event code	The "Event code" field from the AD log (EventCode).
Log event ID	The "Event ID" field from the AD log (EventID).
Log event type	Windows log even type (System/Security/Application etc.)
Log file	Windows log file.

Syslog (Log)

The Syslog log displays events collected by the UserID agent using the "Syslog Sender" connector from syslog source servers. The log displays user logon events and logout events. The following information is displayed:

Name	Description
Node	UserGate node where the event occurred.

Name	Description
Time	The time of the event.
Syslog record details	The link to the event.
Rule	The rule related to the Syslog message.
Severity	Syslog event level.
Object	Detailed information on the process triggering the message (kernel messages, user-level messages, security/authentication etc.).
Computer name	Computer name where the event took place.
Application	Application triggering the event.
Process ID	PID of the process triggering the event.
Data	The event description.

UserID (Log)

The UserID log contains description of events reflecting the result of UserID agent's work. The following information is displayed:

Name	Description
Node	UserGate node where the event occurred.
Time	The time of the event.
Event details	Shows event details.
Action	The action applied to the event.
Log source	The source of the event received.
User	The UG user triggered the event.
IP address	The IP address of the node where the event occurred.
Information	The event description.

RADIUS (Log)

The RADIUS log displays the events collected by the UserID agent from RADIUS accounting data using the RADIUS connector. The log displays user logon events and logout events. The following information is displayed:

Name	Description
Node	UserGate node where the event occurred.
Time	The time of the event.
Rule ID	ID of the rule triggered to cause the event
User	The user, who triggered the event.
Groups	A string of groups the user is a member of.
Status	User status
Source IP	The IP address of the source where the message came from.
NAS IP address	The IP address of the NAS that authorized the user.
User's IP address	User IP address (framed IP address).

Logs Export

The UserGate logs export feature allows you to upload information to external servers for later analysis or SIEM (security information and event management) processing.

UserGate allows you to export the following logs:

- Events
- Web access
- IDPS
- Traffic
- SSH inspection

- DNS
- Mail traffic
- UserID Log.

Sending logs to SSH (SFTP), FTP, and Syslog servers is supported. Logs are sent to SSH and FTP servers according to the schedule specified in the configuration. For Syslog servers, logs are sent immediately after a record is added to the log.

To send logs, you must first create log export configurations in the **Logs export** section.

> **Note**
> **If Log Analyzer is specified in the settings, then processing and export of log files, generating reports and processing of other statistical data are performed by the LogAn server.**

When creating a configuration, provide the following parameters:

Name	Description
Rule name	The name of the log export rule.
Description	Optional field for rule description.
One-time export options	Select the range of log exports. This option is available starting from software version 7.2.0 and up.
Logs to export	Select the log files to export: • Events • Web access • IDPS • Traffic • SSH inspection • DNS • Mail traffic • UserID For each log, you can specify the export syntax: • CEF: Common Event Format (ArcSight) • CEF Compact • JSON: JSON format

Name	Description
	• @CEE: JSON: CEE Log Syntax (CLS) Encoding JSON To select the desired log export format, refer to the documentation for the SIEM system you are using. For a detailed description of log formats, see Description of Log Formats.
Server type	SSH (SFTP), FTP, Syslog.
Server address	IP address or domain name of the server.
Transport	TCP or UDP; applicable only to Syslog servers.
Port	The server port to which the data should be sent.
Protocol	RFC5424 or BSD syslog RFC 3164; applicable only to Syslog servers. Select the protocol compatible with your SIEM system.
Severity	Only for Syslog server type. Optional field; consult the documentation for your SIEM system. Available values: • **Alert**: a state that requires immediate intervention. • **Critical**: a state that requires immediate intervention or signals a fault in the system. • **Errors**: errors detected in the system. • **Warnings**: warnings on potential errors that can occur if no action is taken. • **Notice**: events that relate to unusual system behavior but are not errors. • **Info**: informational messages.
Facility	Only for Syslog server type. Optional field; consult the documentation for your SIEM system. Available values: • **User-level messages** • **System daemon** • **Security/authorization** • **Log audit** • **Log alert** • **Local 0**. • **Local 1**. • **Local 2**. • **Local 3**. • **Local 4**.

Name	Description
	• **Local 5**. • **Local 6**. • **Local 7**.
Hostname	Only for Syslog server type. A unique host name identifying the server that sends data to the Syslog server in the FQDN (Fully Qualified Domain Name) format.
App-Name	Only for Syslog server type. Unique name of the application that sends data to the Syslog server.
Login name	The account name for connecting to the remote server. Not applicable to the Syslog export method.
Password	Account password for connecting to the remote server. Not applicable to the Syslog export method.
Directory path	Server directory to copy log files to. Not applicable to the Syslog export method.
Schedule	Select schedule for sending logs. Not applicable to the Syslog export method. The available options are: • Daily • Weekly • Monthly • Every ... hours • Every ... minutes • Advanced. With the Advanced option, a crontab-like format is used where the date/time string consists of six space-separated fields. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6, where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples:

Name	Description
	"2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".
Manage logs	Manage temporary log files prepared for sending to remote SSH and FTP servers. When sending logs to SSH and FTP servers, UserGate saves the data to send in temporary files. The system copies all files created for sending to a remote server according to the specified schedule. It does not clean up or delete the files. This setting allows you to specify the rotation period for temporary files (in days) or delete any of the temporary files manually. The files are rotated once a day. The system stores a total of N log archives for previous days (according to the number of rotation days) plus one log for the current day.

Data Search and Filtering

Logs normally contain a huge number of records, and not all fields are available in the basic viewing mode. DCFW offers convenient ways to search and filter the information you need. Administrators can search the contents of the logs in basic and advanced modes.

With a simple search, administrators use a graphic interface to set filters by values of the required log fields, thus filtering out unnecessary information. For example, administrators can specify a time range of interest, a list of users, categories, etc. Setting the search criteria is intuitive and does not require any special knowledge.

You can create more complex filters in the advanced search mode using a special query language. In the advanced search mode, you can build queries using log fields that are not available in the basic mode. To construct a query, use field names and values, keywords, and operators. You can enter field values using single or double quotes, or without quotes, if the values do not contain spaces. To group multiple conditions, use parentheses.

Separate keywords by spaces. You can use the following keywords:

Name	Description
AND/and	Logical AND: all query conditions should be met.
OR/or	Logical OR: at least one condition should be met.

The following operators define filter conditions:

Name	Description
=	Equal To. Requires that the field value be completely identical to the specified value. For example, ip=172.16.31.1 displays all log entries where the IP field exactly matches 172.16.31.1.
!=	Not Equal To. Field value must not match the specified value, for example, ip!=172.16.31 displays all log entries where the IP field does not match 172.16.31.1.
<=	Less Than or Equal To. Field value must be less than or equal to the specified value. This can only apply to fields that support comparisons, such as date, portSource, portDest, statusCode, etc., for example, date<='2019-03-28T20:59:59' AND statusCode=303
>=	Greater Than or Equal To. The field value must be greater than or equal to the specified value. This can only apply to fields that support comparisons, such as date, portSource, portDest, statusCode, etc., for example, date>="2019-03-13T21:00:00" AND statusCode=200
<	Less Than. The field value must be less than the specified value. This can only apply to fields that support comparisons, such as date, portSource, portDest, statusCode, etc., for example, date < '2019-03-28T20:59:59' AND statusCode=404
>	Greater Than. The field value must be greater than the specified value. This can only apply to fields that support comparisons, such as date, portSource, portDest, statusCode, etc., for example, (statusCode>200 AND statusCode<300) OR (statusCode=404)
IN	Allows you to specify multiple values for a field in a query. Provide the list of values in parentheses, for example, category IN (botnets, compromised, 'illegal software', 'phishing and fraud','reputation high risk','unknown category')
NOT IN	Allows you to specify multiple values for a field in a query. Displays records that do not contain the specified values. Provide the list of values in parentheses, for example, category NOT IN (botnets, compromised, 'illegal software', 'phishing and fraud','reputation high risk','unknown category')

Name	Description
~	Contains. Allows you to specify a substring that the queried field must contain, for example, browser ~ "Mozilla/5.0" This operator is applicable only to fields that contain string data.
!~	Does Not Contain. Allows you to specify a substring that the queried field must not contain, for example, browser !~ "Mozilla/5.0" This operator is applicable only to fields that contain string data.
MATCH	To specify the substring that must be found in the specified field using the MATCH statement, use JSON format and single quotes, for example, details MATCH '\"module\":\"threats\"' The syntax of queries using this operator is compliant with the RE2 standard. For more details about Google/RE2 syntax, see: https://github.com/google/re2/wiki/Syntax.
NOT MATCH	To specify the substring that must not be found in the specified field using the NOT MATCH statement, use JSON format and single quotes, for example, details NOT MATCH '\"module\":\"threats\"' The syntax of queries using this operator is compliant with the RE2 standard. For more details about Google/RE2 syntax, see: https://github.com/google/re2/wiki/Syntax.

When building an advanced query, DCFW shows possible field names, applicable operators, and possible values, making it easier for the system operator to make complex queries. The list of fields and their possible values for each log may be different.

When you switch from basic to advanced search mode, DCFW automatically generates a search query string that matches the filter specified in the basic search mode.

REPORTS

General Information

Reports allow administrators to provide different slices of data about security events, configurations, or user actions. Reports can be created automatically according to previously created rules and templates and sent to recipients by email.

The **Reports** section contains three subsections: **Templates**, **Report rules**, and **Generated reports**. To create a report, follow these steps:

Name	Description
Step 1. Create a generate report rule.	Create a rule to generate a report and specify all necessary report parameters.
Step 2. Run the report.	Run the report in manual mode or wait until it runs automatically according to the schedule specified in the rule.
Step 3. Receive the report.	Receive the report by mail if you configured the rule to send the report by mail, or download the report from the **Generated reports** section.

Note

Creating a report can take quite a long time and consume a lot of computing resources.

Report Templates

A template defines what the report will look like and what fields it will include. Report templates are provided by the UserGate developer.

Here is the list of report templates by categories:

- **Custom**: a group of templates for generalized statistics of report rule triggering.
- **Captive portal**: a group of templates for events related to user authentication using the Captive portal.
- **Endpoint applications**: a group of templates with lists of applications that were run on the devices.
- **Endpoint rules**: a group of templates for events of endpoint firewall rule triggering.

- **Endpoint events**: shows events received from the devices that are controlled using the UserGate Endpoint software.
- **Events**: a group of templates for events recorded in the event log.
- **IDPS**: a templates group for events recorded in the IDPS log.
- **Network activity**: a templates group for events recorded in the traffic log.
- **Traffic**: a templates group for events recorded in the traffic log and related to the volume of traffic consumed by users, applications, etc.
- **UserID**: a group of templates to create reports on the UserID agent activity.
- **VPN**: a templates group for events related to VPN.
- **Web activity**: a templates group for events recorded in the web access log.

Each template includes a name, report description, and report presentation type (table, histogram, pie).

Report Rules

Report rules set the parameters of the report to be created, as well as the schedule to run the reports and methods of delivering the reports to users. When creating a report rule, administrators specify the following parameters:

Name	Description
Enabled	Enable or disable the report.
Name	The name of the rule.
Description	Optional field for rule description.
Report language	Language to use in the report.
Time range	Time range for preparation of the report.
Report format	Format (PDF, HTML, XML, CSV) of the report. **Important!** Creating reports in PDF results in a high load on the processor and memory. The larger the report, the higher the load. The **Detailed list of all visited URLs** and **Detailed list of all visited sites** reports use CSV format, regardless of the format you select.

Name	Description
Number of records	Set a limit on the number of records displayed in reports that have a limit on the number of top records, for example, the top 20 users who encountered errors authenticating in the web console.
Group by limit (if applicable)	Set a limit on the number of records displayed in reports that have a limit on the number of grouped records, for example, the top 10 users by category: a maximum of 10 users will be listed for each category. This restriction applies only to report templates that contain grouping.
Users	Specify users or user groups for which the report will be created. If not specified, the report will be created for all users.
Templates	List of templates used to build the report. You need to add at least one template.
Schedule	Select a schedule to generate reports. The available options are: • Daily • Weekly • Monthly • Every ... hours • Every ... minutes • Advanced. With the Advanced option, a crontab-like format is used where the date/time string consists of six space-separated fields. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6, where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".
Delivery	

Name	Description
	You can optionally send reports to recipients via the SMTP protocol. To do this, specify the following: • SMTP profile to use for sending reports. For more details about how to configure SMTP profiles, see Notification Profiles. • From: email sender name. • Subject: email subject. • Body: email body. • Recipients: list of the email recipients. The recipients must be added to the lists of the **Emails** library.

Note

Creating a report can take quite a long time and consume a lot of computing resources. It is especially important to consider resource utilization when running reports over a large range of time.

Note

To run a report rule, you do not need to enable it and specify the time when the rule is run. You can manually run any report, including a disabled one, by selecting the rule you want from the list of rules and clicking the Run now button. When created, the report appears under Generated reports.

Generated reports

All generated reports are stored under **Generated reports**. The reports are in PDF or CSV format. For each report the name of the report, which matches the name of the report rule that was used to create this report, the time the report was created, and the size of the report are listed.

To download the report, click the report file, to remove the report, click the **Remove** button.

To customize the storage time of the reports (rotation), click the **Configure** button. The default value is 60 days.

COMMAND LINE INTERFACE (CLI)

GENERAL PROVISIONS

General Provisions (Description)

UserGate DCFW supports command line interface (CLI) management. This interface allows administrators to execute diagnostic and monitoring commands, and to configure, reboot, or shut down devices.

CLI can be useful for troubleshooting network problems or when access to the web console is lost — for example, due to an incorrectly set interface IP address or erroneous zone access control settings that block connections to the web interface.

You can connect to the CLI using the standard VGA/keyboard ports (if physically present on the DCFW equipment), via the serial port, or via SSH over the network.

> **Attention!**
> **If the device has not undergone initial setup, use *Admin* as the login and *usergate* as the password for accessing the CLI.**

The CLI has two operating modes: a diagnostics and monitoring mode, and a configuration mode.

The **diagnostics and monitoring mode** provides commands that allow to view the following information:

- Availability of network resources
- View interface statistics and information
- View ARP entry information
- Perform packet tracing using set rules
- Monitor traffic
- View routing information

- Monitor cluster state
- Monitoring of the IP addresses that were blocked by IDPS.
- Display system information
- Diagnose dynamic routing protocol
- Information about authorized users.

The command line prompt in diagnostic and monitoring mode looks like this:

```
Admin@DCFW>
```

The device is configured in the configuration mode.

To enter the **configuration mode**, use the following command:

```
Admin@DCFW> configure
```

Once you enter the configuration mode, the command line prompt will be as follows:

```
Admin@DCFW#
```

To display a hint on the current possible values, or to complete commands automatically, press the **TAB** key (works in both of the CLI modes). The following symbols can be used in the hint:

***** — a required field in the create command and some others

+ — an optional/variable field

> — a nested field; after entering it the previous list of fields becomes unavailable, a new list of fields appears that can be entered

Example:

```
Admin@nodename# set network virtual-router default
+ interfaces           List of network interfaces attached to this
virtual router
> routes               List of static network routes
```

To connect to the CLI using a monitor and keyboard, follow these steps:

Name	Description
Step 1. Connect a monitor and keyboard to the DCFW device.	Connect a monitor to a VGA (HDMI) port and a keyboard to a USB port.
Step 2. Log in to the CLI.	Log in to the CLI using the login name and password for a user with Full administrator permissions (the default is Admin).

To connect to the CLI using the serial port, follow these steps:

Name	Description
Step 1. Connect to DCFW.	Use a special serial cable or a USB-Serial adapter to connect your computer to DCFW.
Step 2. Launch a terminal.	Launch a terminal that supports serial port connection, such as Putty for Windows or minicom for Linux. Establish a serial port connection using 115200 8n1 as the connection parameters.
Step 3. Log in to the CLI.	Log in to the CLI using the login name and password for a user with Full administrator permissions (the default is Admin).

To connect to the CLI using the SSH protocol, follow these steps:

Name	Description
Step 1. Allow CLI (SSH) access for the selected zone.	Allow SSH access for the CLI protocol in the settings for the zone to which you want to connect for CLI management. The TCP port 2200 will be opened.
Step 2. Launch an SSH terminal.	Launch an SSH terminal on your computer, such as SSH for Linux or Putty for Windows. Specify the DCFW address as the IP address, 2200 as the connection port, and the name of a user with Full administrator permissions as the login name (the default is Admin). For Linux, the connection command should look like this: ssh *Admin@IPDCFW* -p *2200*
Step 3. Log in to the CLI.	Log in to the CLI using the password for the user specified in the previous step.

After successful authorization in the CLI, the user enters the diagnostic and monitoring mode, and a line appears waiting for a command to be entered.

To view the available commands, the current command's possible values, or to complete commands automatically, press the **TAB** key.

To abort the current command, press **Ctrl+C**; to view command history, use the ↑ and ↓ keys.

All CLI commands have the following structure:

```
<action> <level> <filter> <configuration_info>
```

where:

<action> is the action to be performed;

<level>: a configuration level; the levels correspond to the DCFW web interface sections.

<filter> is the identifier of the object being accessed; and

<configuration_info> is the set of parameter values to be applied to the <filter> object.

CLI supports multi-line command entry. To move to a new line, add "\" at the end of the current one. Starting from the second line, entering "\" is not required; to finish the entry, enter one empty line:

```
Admin@nodename# set users user example \
... name username1
... enabled on
... groups [ "Default Group" ]
...
Admin@nodename#
```

COMMANDS AVAILABLE PRIOR TO INITIAL NODE SETUP

Commands Available Prior to Initial Node Setup (Description)

If the device has not undergone initial configuration, diagnostics and monitoring commands are fully available in the CLI, but only network configuration commands are available in the configuration mode (zone, interface, gateway, and virtual router configuration as well as enabling/disabling remote access to the radmin-emergency server).

Available Commands in Diagnostic Mode

The diagnostics commands can be used to:

- Availability of network resources
- View interface statistics and information
- View ARP entry information
- Monitor traffic
- View routing information
- Display system information
- Diagnose dynamic routing protocol

For more information on the syntax and examples of using diagnostic commands, see the Diagnostics and Monitoring Commands section.

Available Commands in Configuration Mode

To enter the configuration mode, use the following command:

```
Admin@NGFW> configure
```

Once you enter the configuration mode, the command line prompt will be as follows:

```
Admin@NGFW#
```

In configuration mode, the following commands are available: *execute* (commands that are not related to device configuration: ping, date, traceroute, etc.), network

configuration commands (configuring zones, interfaces, gateways, and virtual routers), and commands for enabling/disabling remote access to the radmin-emergency server.

For more information on the syntax and examples of using the *execute* commands, see the Configuration Mode section.

For more information on the syntax and usage examples of network configuration commands, see the following sections:

- Zone Configuration Commands
- Interface Configuration Commands
- Gateway Configuration Commands
- Virtual Router Configuration Commands

For more information on the syntax and examples of using commands to enable/disable remote access to the radmin-emergency server, see the Configuring Device Management section.

For more information about initial device provisioning using the CLI, see the Initial Setup section.

INITIAL SETUP

Initial Setup (Description)

There are several ways to perform the first initialization of DCFW using the CLI.

Install UserGate as the master node.

To set DCFW as a master node, use the following command:

```
Admin@nodename# execute install master
```

Specify the following parameters:

Parameter	Description
login	Set admin name.
password	Set a password for the administrator account. You can also set the password on pressing **Enter** after typing in the administrator login; the password must be entered twice.

Install UserGate as a slave node.

To set DCFW as an additional cluster node, use the following command:

```
Admin@nodename# execute install slave
```

Specify the following parameters:

Parameter	Description
interface	The interface for connecting to the cluster.
slave-ip	The IP address that will be assigned to the interface used for connecting to the cluster.
gateway-address	Gateway IP address. A gateway is required if the nodes are in different subnets.
master-ip	The master node IP address.
master-secret	The master node secret used to connect the node to the cluster.
login	A DCFW administrator's login.
password	The password for the administrator account.

Configuration using UserGate Management Center.

To configure DCFW using UGMC, use the following command:

```
Admin@nodename# execute install mc
```

Specify the following parameters:

Parameter	Description
login	A DCFW administrator's login.
password	The password for the administrator account.
mc-ip	UGMC server IP address.
device-code	The unique device code used for connecting the node to UGMC.

After the initial setup, the full management functionality will be available from the CLI.

DIAGNOSTICS AND MONITORING COMMANDS

Diagnostics and Monitoring Commands (Description)

The diagnostics commands can be used to:

- Availability of network resources
- View interface statistics and information
- View ARP entry information
- Perform packet tracing using set rules
- Monitor traffic
- View routing information
- Monitor cluster state
- Monitor IDPS-blocked IP addresses
- Display system information
- Diagnose dynamic routing protocol

- Information about authorized users.

Basic control commands

To view the current date and time on the node, use the command:

```
Admin@nodename> date
```

To reboot the node, use the command:

```
Admin@nodename> reboot
```

To shut down a node, use the command:

```
Admin@nodename> shutdown
```

To switch to the node configuration mode, use the following command:

```
Admin@nodename> configure
```

To exit the CLI, use the following command:

```
Admin@nodename> exit
```

Commands for Checking the Availability of Network Resources

To check the availability of a specific host using the ping utility, use the command:

```
Admin@nodename> ping <parameters>
```

The following parameters can be used with the command:

Parameter	Description
host	An IP address or domain name of the host.

Parameter	Description
count	The number of echo requests to send. If not specified, the system will send the packets until the user terminates the connection (to terminate sending, press Ctrl+C).
interface	The address of the selected interface will be used as the source address for running ping.
interval	The time between sent packets (in seconds).
mtu	The MTU size of the sent packets.
numeric	Don't resolve names.
ttl	The packet's time to live.
timestamp	Display timestamps.
virtual-router	Name of the virtual router.

To trace a connection to a specific host, use the following command:

```
Admin@nodename> traceroute <parameters>
```

The following parameters can be used with the command:

Parameter	Description
host	An IP address or domain name of the host being traced.
interface	The interface from which packets will be sent.
min-interval	Minimum interval between packets.
not-map-ip	Do not search the hostname for the IP address when displaying.
port	Specify a port instead of the default port (1-65535).
use-icmp-echo	Use ICMP echo.

To check the availability of a third-party HTTP/HTTPS server, use the following command:

```
Admin@nodename> netcheck <parameters>
```

The following parameters can be used with the command:

Parameter	Description
address	The host's domain name for checking availability over TCP or an URL for HTTP.
type	Check availability over: • **http** • **tcp** (if no port is specified, then port 80 is used by default).
check-cert	Check SSL certificate.
dns-ip	A DNS server's IP address.
dns-tcp	Use TCP instead of UDP for DNS request.
data	Request the site content. Only headers are requested by default.
timeout	Maximum timeout for a server response.
useragent	A parameter for specifying the browser type (useragent). Some sites may permit access from certain browsers only. The parameter value is specified in double quotes.

To check the DNS record of a domain, the following command is used:

```
Admin@nodename> dig <parameters>
```

The following parameters can be used with the command:

Parameter	Description
host	A host's domain name or IP address for reverse lookup.
dns	Specify the IP address of the DNS server.
reverse-lookup	Getting hostname from IP address.
tcp	Use TCP instead of UDP.

To check the IP address ownership against the current GeoIP database, use the following command:

```
Admin@nodename> check-geoip ip <IP-address>
```

Interface Statistics and Information

To display interface information, use the following command:

```
Admin@nodename> show network interface
```

To display the statistics and information for a specific interface, use the following command:

```
Admin@nodename> show network interface <interface-name>
```

You can also choose to display only the information or statistics:

```
Admin@nodename> show network interface <interface-name> type info
Admin@nodename> show network interface <interface-name> type statistics
```

To display the ordered list of network interface names and their corresponding physical addresses, use the following command:

```
Admin@nodename> show network interfac-mapping
```

The interfaces are ordered b port number on the PCI bus.

To delete the list, use the following command:

```
Admin@nodename> clear network interfac-mapping
```

After the UserGate device reboots, the list will update and become available for display. This operation needs to be performed after adding network ports to a configured UserGate device.

ARP Entries

To view ARP entry information, use the following command:

```
Admin@nodename> show network arp
```

You can filter the displayed entries using these filtering options:

Parameter	Description
node-name	The name of the cluster node whose ARP entries need to be displayed. Next, specify the interface name or host IP address: `Admin@nodename> show network arp node-name <node-name> interface <iface-name>` `Admin@nodename> show network arp node-name <node-name> host <ip>`
interface	The name of a DCFW interface.
host	The IP address of the device.
mac	The MAC address of the device.

```
Admin@nodename> show network arp host <IP-address>
Admin@nodename> show network arp interface <interface-name>
Admin@nodename> show network arp mac <MAC-address>
```

You can also view ARP entries in the configuration mode. The commands are identical to those used in the diagnostics and monitoring mode.

Note

The diagnostics and monitoring mode provides actions relating to system ARP entries, while the configuration mode deals with static entries.

Static ARP entries can be added in the configuration mode using the following command:

```
Admin@nodename# set network arp host <IP-address> interface <interface-name> mac <MAC-address>
```

Command parameters:

Parameter	Description
node-name	The name of the cluster node on which the ARP entry will be created. Next, specify the interface name and the IP and MAC addresses of the device.
interface	The name of a DCFW interface.
host	The IP address of the device.
mac	The MAC address of the device.

The commands for deleting system and static ARP entries have a similar structure and differ only in the action to be taken:

- **clear**: delete system records in the diagnostics and monitoring mode
- **delete**: delete static records in the configuration mode.

The format of the deletion commands is shown below using diagnostics and monitoring commands as an example.

To delete a system entry:

```
Admin@nodename> clear network arp interface <iface-name> host <ip>
```

To delete an entry on a different cluster node:

```
Admin@nodename> clear network arp interface <iface-name> node-name <node-name> host <ip>
```

The following command deletes all system records on the specified interface(s):

```
 Admin@nodename> clear network arp interfaces [ <iface-name1> <iface-
name2> … ]
```

To delete all system entries for an interface on a different node:

```
 Admin@nodename> clear network arp interfaces [ <iface-name1> <iface-
name2> … ] node-name <node-name>
```

Packet Tracing

To perform packet tracing, use the following command:

```
Admin@nodename> show network trace
```

It will display information such as the source and destination IP addresses, protocol, UserGate source and destination port names, and source and destination TCP/UDP port numbers. This command is also available in the configuration mode.

To exit the packet tracing mode, press **Ctrl+C**.

Packet tracing rules are created and configured in the configuration mode at the **network** level. To create a rule, use the following command:

```
Admin@nodename# create network trace-rules
```

Next, specify the following parameters:

Parameter	Description
enabled	Enable or disable the packet tracing rule: • **on** • **off**
name	The name of the rule. If not set, the name is generated automatically as trace_rule_N, where N is the ordinal number of the packet tracing rule being created.
zones-in	The list of traffic source zones.

Parameter	Description
source-ip-lists	The list of source IP address groups for the packets. For more details on creating IP address groups using the CLI, see the section Configuring IP Addresses.
source-ip-addresses	The list of source IP addresses for the packets.
dest-ip-lists	The list of destination IP address groups for the packets. For more details on creating IP address groups using the CLI, see the section Configuring IP Addresses.
dest-ip-addresses	The list of destination IP addresses for the packets.
services	Service type. For more details, see the Configuring Services section.

Example command to create a rule:

```
Admin@nodename# create network trace-rules enabled on name "Test trace"
source-ip-addresses [ 192.168.0.100 ]
```

Example command to edit a rule:

```
Admin@nodename# set network trace-rules <trace-rule-name>

Admin@nodename# set network trace-rules "Test trace" services
[ "[SYSTEM] Any ICMP" ]
```

All the parameters listed in the table above can be modified.

To view the existing packet tracing rules:

```
Admin@nodename# show network trace-rules
```

To delete a packet tracing rule, use the following command:

```
Admin@nodename# delete network trace-rules <trace-rule-name>
```

The values of individual rule parameters can also be deleted. These are available for deletion:

- **zones-in**
- **source-ip-lists**
- **source-ip-addresses**
- **dest-ip-lists**
- **dest-ip-addresses**
- **services**

Traffic Monitoring

To monitor traffic, use the following command:

```
Admin@nodename> show traffic
```

Parameter	Description
flows	Displays information about the incoming and outgoing flows. Filtering is available by: • **source-ip**: the source IP address • **source-port**: the source port • **dest-ip**: the destination IP address • **dest-port**: the destination port • **vlan-tag**: the VLAN tag. • **interface-name**: the name of the interface • **node-name**: the node name • **protocol**: the protocol
connections	Displays information on connections (the protocol and its number, record TTL; source and destination IP addresses, source and destination ports; source and destination IP addresses, source and destination ports expected in the response; session status (UNREPLIED or ASSURED); number of sent and received packets and bytes; source zone; whether this is a session of a known DCFW user; etc.). Filtering is available by: • **protocol**: the protocol

Parameter	Description
	• **source-ip**: the source IP address • **dest-ip**: the destination IP address • **node-name**: the node name • **expect**: display non-established connections. The options are: ◦ **on** ◦ **off**
capture	Displays packet capture. Filtering by the following parameters is available: • **destination**: the destination IP address • **destination-port**: the destination port • **ipv4-protocol**: the IPv4 protocol number (0-255) • **interfaces**: the name of the interface • **protocol**: select a protocol • **rule**: select an existing rule for packet capture • **source**: the source IP address • **source-port**: the source port

Example traffic monitoring command:

```
Admin@nodename> show traffic connections node-name utmcore@dineanoulwer
dest-ip 192.168.0.100 expect on
```

LLDP

To view the information received via LLDP (Link Layer Discovery Protocol), use the following commands:

```
Admin@nodename> show lldp
Admin@nodename> show lldp neighbors
Admin@nodename> show lldp statistics
```

Command parameters:

Parameter	Description
neighbors	The list of LLDP-compatible devices with LLDP advertisement enabled. • **Chassis ID**: the chassis ID • **SysName**: the name of the system • **SysDescr**: a description of the system containing information on the device's hardware and operating system • **Management**: the neighboring device address (contains IPv4 and IPv6 addresses and interface number for the specified management address) • **Capability**: the device's function (e.g., router, switch, etc.) • **Port ID**: the ID of the port from which the LLDPDU (Link Layer Discovery Protocol Data Unit) was transmitted • **PortDescr**: a description of the port • **TTL**: the TTL of the transmitted LLDP packets
statistics	The statistics on the interfaces for which a LLDP profile was specified: • **Interface**: the name of the Interface • **Transmitted**: the total LLDP frame count transmitted via the interface. • **Received**: the total LLDP frame count received on the interface. • **Discarded**: the number of LLDP frames received on this interface that were discarded. • **Unrecognized**: the number of LLDP frames with unconfirmed content received on this interface. • **Ageout**: each LLDP frame contains information on how long the LLDP information is valid (the ageout). If no new frames are accepted during the ageout period, the LLDP information is deleted. • **Inserted**: the number of added records containing information on LLDP neighbors. • **Deleted**: the number of deleted records containing information on LLDP neighbors.

> **Note**
> **To be able to view the information received via LLDP, the LLDP service must be activated on DCFW (LLDP profiles configured in the item library and activated in the interface settings).**

Routes

This section allows you to perform diagnostics and monitoring for route information on DCFW.

To view all routes contained in the default router, use the following command:

```
Admin@nodename> show network route
```

Parameter	Description
ip	IP address to which you want to display the route.
node-name	Select a cluster node.
connected	Routes to networks connected directly to DCFW interfaces. These routes are marked with a **C** in the route list.
kernel	Display the routes added by the administrator. These routes are marked with a **K** in the route list.
summary	Number of active connections and FIB (Forwarding Information Base) records.
ospf	Display routes received using the OSPF dynamic routing protocol. These routes are marked with a **O** in the route list.
bgp	Display the routes received using the BGP dynamic routing protocol. These routes are marked with an **B** in the route list.
rip	Display the routes received using the RIP dynamic routing protocol. These routes are marked with an **R** in the route list.
virtual-router	Virtual router for which you want to display routes (<vrf-name> | **all**).

OSPF Monitoring

To diagnose and monitor OSPF, use the following commands. Display OSPF information:

```
Admin@nodename> show network ospf
...
Admin@nodename> show network ospf <parameter>
```

Parameter	Description
node-name	Select a cluster node.
virtual-router	Virtual router for which you want to preview general OSPF information: (<vrf-name> | **all**).
route	Display routes received using the OSPF dynamic routing protocol.
database	Display the following information: • **Router Link States**: routers use Type 1 Link State Advertisement (LSA) (Router LSA) packets to send information within the same zone; they are used to transmit information about their own and their neighbors' interfaces to their neighboring routers in the same zone. • **Network Link States**: a Designated Router (DR) generates LSA Type 2 (Network LSA) packets to describe all routers connected directly to its segment. • **Summary Link States**: Area Border Routers (ABR) generate LSA Type 3 (Summary LSA) packets. These packets contain summary messages about the directly connected zone, report information to other zones to which the ABR is connected, and are transmitted to multiple zones throughout the network. • **ASBR-Summary Link States**: LSA Type 4 (ASBR Summary LSA) packets report the presence of an Autonomous System Border Router (ASBR) in other areas.
neighbor	Display information on neighbors: • Neighbor ID (router ID). • Priority. The router with the highest priority becomes the Designated Router, DR. If router priorities are equal, the router with the highest ID will be selected. • Status, such as Full/DR, Full/BDR, Full/Drother.

Parameter	Description
	• Idle interval: the time interval before the connection to the OSPF neighbor is terminated if no Hello packet has been received. • IP address of the interface to which the neighbor is connected. • Interface on which the router adjacency is formed. Additional parameters: • **interface-name**: display neighbors with which adjacency is established on the specified interface • **all**: display the table with all neighbors. • **detail**: display detailed information about neighbors.
interface	Display OSPF interface information. Additional parameters: • **interface-name**: display information about the specified interface • **traffic**: display the statistics of transmitted and received OSPF packets (Hello, Database Description, Link State Request, Link State Update, Link State Acknowledgment).
border-routers	Display information about border routers.

Restart the OSPF process:

```
Admin@nodename> clear network ospf <parameter>
```

Parameter	Description
interface-name	The interface name.
node-name	Select a cluster node.
virtual-router	Virtual router on which you want to restart OSPF (<vrf-name> | **all**).
interface	Interface on which you want to restart the OSPF process (<interface-name>).
neighbor	Select neighbors for which the process will be restarted.

BGP Monitoring

To diagnose and monitor BGP, use the following commands.

Display the router's BGP table:

```
Admin@nodename> show network bgp
...
Admin@nodename> show network bgp <parameter>
```

Parameter	Description
node-name	Select a cluster node.
virtual-router	Virtual router for which you want to display routes (<vrf-name> | **all**).
ip	IP address to which you want to display the route.
statistics	Display BGP statistics.
neighbors	Display information about BGP neighbors (to display information about a specific neighbor, provide its IP address). Additional parameters available to use to specify a neighbor: • **received-routes**: the routes received before the incoming policy is applied to them (Routemap and filters) • **advertised-routes**: the routes advertised to the specified neighbor.
summary	Display summary information on neighbors.

Re-request information from BGP neighbors (TCP session break):

```
Admin@nodename> clear network bgp
```

Available parameters:

Parameter	Description
ip	IP address of the neighbor to which the connection will be interrupted to update information.

Parameter	Description
node-name	Select a cluster node.
virtual-router	Name of the virtual router to which the BGP neighbor belongs.

In case the neighbor devices support the Route Refresh method you can send a special message like ROUTE REFRESH instead of reinitializing the entire session with the neighbor. You can send this message to update information without interrupting the routing.

To update information without interrupting the session with the neighbor, use the following command:

```
Admin@nodename> clear network bgp ip <neighbor-ip> soft in | out
Admin@nodename> clear network bgp virtual-router <vrf-name> ip
<neighbor-ip> soft in | out
```

RIP Monitoring

To diagnose and monitor RIP, use the following commands.

Display RIP information from the default router table (network address received via RIP, Next Hop address, route metric, route tag to separate internal and external routes, and timeout to invalidate the route if no information about it has been received):

```
Admin@nodename> show network rip
...
Admin@nodename> show network rip <parameter>
```

Additional available parameters:

Parameter	Description
node-name	Select a cluster node.
status	Current RIP status: version, timers, filters, routes distributed, etc.
virtual-router	Virtual router for which you want to preview RIP route information: <vrf-name> | **all**.

Multicast traffic monitoring

To view the multicast traffic routing table on the default router, use the following command:

```
Admin@nodename> show network mroute
...
Admin@nodename> show network mroute <parameter>
```

Additional available parameters:

Parameter	Description
node-name	Select a cluster node.
count	Display statistics about the group and the source.
virtual-router	Select a virtual router: <vrf-name> | **all**.
summary	Summary of each record in the multicast routing table.
fill	Multicast traffic routing table. Additional parameter: • **ip**: display the entry for a particular IP address (the IP address should follow).
ip	Display the record for a particular IP address (provide the IP address).

IGMP Monitoring

To monitor IGMP (Internet Group Management Protocol) operation, use the following command (the parameters are required). Display information for the default router:

```
Admin@nodename> show network igmp <parameters>
```

Parameters:

Parameter	Description
node-name	Select a cluster node.
virtual-router	Select a virtual router.

Parameter	Description
statistics	Message statistics: • IGMP Membership Query is a message from the server to the client with a request to renew the client's group subscriptions lest the server stop broadcasting the group(s) to this network segment. • IGMP Leave is a message from the client to the server notifying that the client wants to remove the multicast group from the list of group subscriptions. • IGMP Membership Report is a message from the client to the server notifying that the client wants to receive this group's traffic.
join	Display information about IGMP groups.
sources	Display information about multicast traffic sources.
groups	Display the multicast groups received via IGMP protocol. The following information is displayed: • Total number of groups. • Interface via which the group is available. • Group address. • INCLUDE or EXCLUDE mode. • Timer that determines the period for which the router will stop forwarding traffic to the interface if no IGMP Membership Report has been received. • How long the group is known.
interface	Display the interface information related to multicast routing: • Interface name, status, and address. • IGMP version. • Querier and its address. • Timer that is reset every time a Query message with a lower IP address arrives. You can specify: • **interface-name**: the name of the interface • **detail**: detailed information about the interface.

PIM Monitoring

To monitor PIM (Protocol-Independent Multicast), use the following command (the parameters are required). Display information for the default router:

```
Admin@ndename> show network pim <parameter>
```

Parameters:

Parameter	Description
node-name	Select the cluster node for which you want to preview information.
virtual-router	Select the virtual router for which you want to preview information.
vxlan-groups	Information about VXLAN groups used in multicast.
statistics	Protocol statistics.
join	Display information about PIM groups.
neighbor	Information about the neighbors: • Interface via which the neighbor information was obtained. • Neighbor's address. • Time since PIM was last started. • How long the neighbor is available. • DR priority.
next-hop	Records about the next-hop addresses.
state	Information about known S and G routes, IIF (Incoming Interface), and OIL (Outgoing Interface List).
rp-info	Display information about the Rendezvous Point (RP), such as the address and allowed ASM groups from this RP.
interface	Information about interfaces configured for PIM, such as the interface name and address, DR address, etc. Additional parameters: • **interface-name**: the name of the interface • **traffic**: the sent/received message statistics

Parameter	Description
	• **detail**: detailed information about the interface.
group-type	List of allowed group addresses for SSM (Source Specific Multicast).
secondary	Display information about the interface by specifying an additional IP address.

Cluster State Monitoring

Cluster state monitoring commands can be run on any of the nodes in the cluster. They allow getting information about the current state of the cluster, its nodes, operation mode and state transition history.

To monitor the state of the cluster as a whole, use the following command:

```
Admin@nodename> show ha-cluster state
```

To monitor the state of cluster nodes, use the following command:

```
Admin@nodename> show ha-cluster tablestat
```

To display the state transition history for the cluster, use the following command:

```
Admin@nodename> show ha-cluster failover
```

Monitoring IDPS-Blocked IP Addresses

To view the table of IDPS-blocked IP addresses, use the following command:

```
Admin@nodename> show blocked-ip
```

To unblock individual IP addresses, use the following command:

```
Admin@nodename> clear blocked-ip ips [ ip-address ip-address ... ]
```

Display System Information

To view the software version of the system, use the following command:

```
Admin@nodename> show system version
```

To display information on the number of active TCP/UDP/ICMP sessions in the system, use the following command:

```
Admin@nodename> show system sessions
```

To display information on the number of active sessions by protocol or time interval, use the following command:

```
Admin@nodename> show system sessions counters [ parameters ]
```

Clear statistics:

```
Admin@nodename> clear system sessions
```

Dynamic Routing Protocol Diagnostics

The commands in this section can be used to view event entries in the debug logs for dynamic routing protocols. Events of a specific protocol are included in the debug log using the debug command in configuration mode (read more in the Configuration Mode section).

To view entries in a debug log, use the following command:

```
Admin@nodename> show log routing <parameters>
```

One of the parameters below should follow:

Parameter	Description
all	All protocols.
rip	RIP protocol.

Parameter	Description
bgp	BGP protocol
igmp	IGMP protocol
pim	PIM protocol.
ospf	OSPF protocol
bfd	BFD protocol
msdp	MSDP protocol
mroute	The mroute multicast route table.
ssmpingd	The ssmpingd multicast testing tool.

To output the events from a debug log to the console in real time, use the following command:

```
Admin@nodename> show log tail on routing <parameters>
```

Specify one of the parameters from the parameter table for the debug log entry view command shown above.

To disable event output from a debug log to the console in real time by protocol, use the following command:

```
Admin@nodename> show log tail off routing <parameters>
```

The parameters are similar to those described above.

Viewing Information About Authorized Users

To view information about all authorized users, use the following command line interface command:

```
Admin@nodename> show user-auth
```

To view the details of an authentication session for a specific user, use the command:

```
Admin@nodename> show user-auth <parameter>
```

Either the username (login) or the IP address (ip-address) can be used as the parameter.

To delete a session of a specific user, use the command:

```
Admin@nodename> clear user-auth <parameter>
```

Either the username (login) or the IP address (ip-address) can be used as the parameter.

CONFIGURATION MODE

Configuration Mode

To enter the configuration mode, use the following command:

```
Admin@nodename> configure
```

Once you enter the configuration mode, the command line will be as follows:

```
Admin@nodename#
```

To view a hint about the current possible values or to autocomplete commands, press the **Tab** key. The following symbols can be used in the hint:

***** — a required field in the create command and some others

+ — an optional/variable field

> — a nested field; after entering it the previous list of fields becomes unavailable, a new list of fields appears that can be entered

Example:

```
Admin@nodename# set network virtual-router default
* name                  Name
+ description           Description
+ interfaces            List of network interfaces attached to this
virtual router
> bgp                   BGP router
> multicast-router      Multicast router
> ospf                  OSPF router
> rip                   RIP router
> routes                List of static network routes
```

General Command Structure in Configuration Mode

CLI commands have the following structure:

```
<action> <level> <filter> <configuration_info>
```

where:

<action> is the action to be performed;

<level>: a configuration level; the levels correspond to the DCFW web interface sections.

<filter> is the identifier of the object being accessed; and

<configuration_info> is the set of parameter values to be applied to the <filter> object.

Name	Description
<action>	The following actions are available in the configuration mode: • **execute**: execute commands not related to UserGate configuration (ping, date, traceroute, etc.). The command is available regardless of the configuration level (<level>). • **set**: edit all objects and enable various parameters, e.g. radmin.

Name	Description
	• **end**: go one level up. • **show**: display the current values. You can use this at any configuration level. Displays everything below the current level. • **edit**: go to a specific configuration level. The configuration level is displayed under the command line. • **top**: go back to the topmost configuration level. • **exit**: exit the configuration mode. • **export**: export the configuration. • **import**: import the configuration. • **create**: create new objects. • **delete**: delete an object or a parameter from the parameter list. • **debug**: enable logging of dynamic routing protocol events. For example, to view information about all interfaces, run the following command: `Admin@nodename# show network interface` To go to the **network interface** level, use the following command. The current level will be displayed above the command line: `Admin@nodename# edit network interface` `[ network interface ]` `Admin@nodename#` After you go to the **network interface** level, use the show command to show all interfaces without specifying a level: `Admin@nodename# show` `adapter:` `  port0` `  interface-name  : port0` `  node-name  :` `utmcore@dineanoulwer`

Name	Description
	zone : Management enabled : on ip-addresses : 192.168.56.3/24 iface-mode : dhcp ... ... To return from the **network interface** level back to the general level of the configuration mode, use the **end** command twice: [network interface] Admin@nodename# end [network] Admin@nodename# end Admin@nodename# To return to the topmost level of the configuration with a single command, you can use the **top** command: [network interface] Admin@nodename# top Admin@nodename#
<level>	Levels in the command line follow the web interface of UserGate DCFW: • **security-policy**: corresponds to the **Security policies** section of the web interface. • **network**: corresponds to the **Network** section of the web interface. • **settings**: corresponds to the **UserGate** section of the web interface. • **global-portal**: corresponds to the **Global portal** section of the web interface. • **network-policy**: corresponds to the **Network policies** section of the web interface. • **vpn**: corresponds to the **VPN** section of the web interface.

Name	Description
	• **users**: corresponds to the **Users and devices** section of the web interface. • **libraries**: corresponds to the **Libraries** section of the web interface. • **monitoring**: corresponds to the **Diagnostics and monitoring** section of the web interface. • **waf**: corresponds to the **WAF** section of the web interface.
<filter>	ID of the object which is being accessed. Objects are identified by their name. If there are objects with identical names or it is more convenient to identify objects by another parameter, specify <configuration_info> in parentheses (this is discussed later in the section). This will find an object matching all the fields specified in parentheses. For example, you need to display information about the port0 interface on another cluster node. The command `Admin@nodename# show network interface adapter port0` will display information about the interface port0 on the current UserGate node. To preview information about the port0 interface on another node (named another_node for instance), you need to explicitly specify the node name in parentheses: `Admin@nodename# show network interface adapter ( node-name another_nodename interface port0 )` **Important!** Parentheses should be separated by spaces on both sides.
<configuration_info>	Set of parameter-argument pairs. where the parameter is the name of the field for which you need to set the argument. Arguments can be single-valued or multi-valued. **A single-valued argument** is the value of the parameter. If the string contains spaces, use quotation marks. For example, to create a group named VPN users: `Admin@nodename# create users group "VPN users"`

Name	Description
	Multi-valued arguments are used to set multiple values of a parameter; include them in square brackets and separate by spaces. For example, you want to add user1 and user2 to the "VPN users" group. Then you need to set [user1 user2] as the argument for the **users** parameter: `Admin@nodename# set users group "VPN users" users [ user1 user2 ]` **Important!** Square brackets should be separated by spaces on both sides.

Execute Commands

These commands have the following structure:

```
Admin@nodename# execute <command-name>
```

Available commands:

Parameter	Description
update	Update: • **software-updates**: software update • **libraries-updates**: library update. You can update all libraries at once or individual libraries.
traceroute	Traceroute the connection to a specified host. Available parameters: • **host <ip-or-domain>**: the IP address or the name of a domain being traced. • **interface <iface-name>**: the interface from which packets will be sent • **not-map-ip**: do not search the hostname for the IP address when displaying • **use-icmp-echo**: use ICMP echo. • **port**: specify a port instead of the default port (1-65535). • **min-interval**: minimum interval between packets.

Parameter	Description
	`Admin@nodename# execute traceroute hostname <hostname>`
license	The product registration command has the following structure: `Admin@nodename# execute license activate <pin-code>` Provide your product activation code a <pin-code>.
logs	• **send-once**: a rule-based one-time log sending command.
termination	Close the administrator sessions. For more details, see Configuring administrator sessions.
cache	Clear LDAP record cache: • **ldap-clear**.
check-geoip	Checking the ownership of an IP address using the current GeoIP database.
ping	Ping a specific host. Available parameters: • **host**: the host's IP address or domain name. • **count**: the number of echo requests to send. If not specified, the system will send the packets until the user terminates the connection (to terminate sending, press Ctrl+C). • **numeric**: do not resolve names. • **timestamp**: display timestamps. • **interval**: the time between sent packets (in seconds). • **ttl**: the packet's time to live. • **interface**: the address of the selected interface will be used as the source address for running ping. • **mtu**: the MTU size of the sent packets. • **virtual-router**: virtual router name. `Admin@nodename# execute ping hostname <hostname> count <number>`

Parameter	Description
reboot	Reboot the UserGate server.
date	View the current date and time on the server.
shutdown	Shutting down the UserGate server.
netcheck	Check the availability of a third-party HTTP/HTTPS server. You can use the following parameters: • **address**: the host's domain name for checking availability over TCP or URL for HTTP • **dns-ip**: the DNS server's IP address • **dns-tcp**: use TCP instead of UDP for DNS request • **check-cert**: check the SSL certificate • **type**: check availability over: ◦ **http** ◦ **tcp** (if no port is specified, port 80 is used by default). • **data**: request the site content. Only headers are requested by default. • **timeout**: the maximum time to wait for a reply from the web server. • **user-agent**: parameter to specify the browser type (useragent). Some sites may only allow access from certain browsers. The parameter value is specified in double quotes. `Admin@nodename# execute netcheck type tcp address <host-domain-name> data on` `Admin@nodename# execute netcheck address <host-domain-name>`
dig	Check the domain DNS record. • **host**: the host's domain name or an IP address for reverse lookup. • **reverse-lookup**: get the host from the IP address • **dns**: specify the IP address of the DNS server • **tcp**: use TCP instead of UDP.

Parameter	Description
	`Admin@nodename# execute dig hostname <host-domain-name>` `Admin@nodename# execute dig hostname <IP-address> reverse-lookup on`
configurate-cluster	Generate the secret code required for adding a new node to the configuration cluster: `Admin@nodename# execute configurate-cluster generate-secret-key <parameter>` • **secret**: the key for secret code generation in the [0-9a-zA-Z]+#[0-9a-zA-Z]+ format (e.g., example#key) • **expiration-time**: the expiration time of the code in seconds • **request-limit**: the validity time of the code generation request **Important!** Using this command requires a **Cluster** module license, otherwise an error message will be displayed.
mc-force-disconnect	The command for an emergency disconnection of the node from the MC with which it was integrated. Depending on the command's argument, the objects imported from MC are saved locally or deleted: • **keep**: disconnect from MC and keep all imported objects (libraries, rules, etc.). The objects imported from MC are converted to local ones. • **delete**: disconnect from MC and delete all imported objects (libraries, rules, etc.). The imported objects that are currently in use are converted to local ones. `Admin@nodename# execute mc-force-disconnect keep` `Admin@nodename# execute mc-force-disconnect delete`
firewall	

Parameter	Description
	Firewall operations: • **force-changes**: reapply all firewall rules and terminate current sessions.
restore-mac	Restore the MAC address of the interface.

Some of the commands listed above, except for product update, product registration, administrator session management, and cache clearing are also available in the diagnostics and monitoring mode. To execute them, use the following command:

```
Admin@nodename> <command-name>
```

Import Commands

Import is available in the sections **Settings**, **Users**, **Network**, **Network policies**, **Security policies** , **Global portal**, **VPN**, <0>WAF.

You can import a certificate under UserGate settings. For more details, see the Configuring Certificates section.

In the sections **Users**, **Network**, **Network policies**, **Security policies**, **Global portal**, **VPN, WAF** Import of rules written in UPL is available. When you import rules, they replace the existing rules. You can specify multiple rules at once.

Under **Users**, you can import Captive portal rules. For more details on adding rules, see the respective subsection of the Captive Portal Configuration section.

Under **Network**, you can import DNS rules. For more details, see the Configuring DNS Rules section.

Under **Network policies**, you can import the rules for the firewall, NAT and routing, bandwidth management, and load balancing. For more details, see Configuring Firewall Rules, Configuring NAT and Routing Rules, and Configuring Traffic Shaping Rules. Configuring load balancing.

Under **VPN**, you can import server and client rules. For details on adding these rules, see Configuring Server Rules and Configuring Client Rules.

Export Commands

You can export certificates and library items.

For more details on exporting certificates, see Configuring Certificates.

You can export the following library items: IP addresses, useragents, URL lists, URL categories, overridden URL categories, content types, morphology, emails, and phone numbers. To export a library item, use the following command:

```
Admin@nodename# export libraries <library-name> <list-name>
```

where:

<library-name> is the item library name (IP addresses, URL lists, etc.), and

<list-name> is the library item name.

Debug Command

The debug command allows you to enable logging of routing protocol events. Events are recorded in the debug log. They can also be viewed in monitoring mode in the CLI console (for more details, see the Diagnostics and Monitoring section).

To enable logging of a specific routing protocol, use the command:

```
Admin@nodename# debug <protocol> <parameters>
```

The following protocols are supported:

Parameter	Description
rip	RIP protocol.
bgp	BGP protocol
igmp	IGMP protocol
pim	PIM protocol.
ospf	OSPF protocol
bfd	BFD protocol

Parameter	Description
msdp	MSDP protocol
mroute	The mroute multicast route table.
ssmpingd	The ssmpingd multicast testing tool.

DEVICE SETUP

Device Setup (Description)

Configuring CLI

You configure the command line interface at the **settings cli** level. To configure the level of diagnostic details, use the following command:

```
Admin@nodename# set settings cli log-level <off | error | debug |
warning | info>
```

Detail levels:

- **off**: disable logging
- **error**: errors only
- **debug**: maximum level of detail
- **warning**: errors and warnings
- **info**: errors, warnings, and additional information.

To display CLI settings, use the following command:

```
Admin@nodename# show settings cli
```

To configure the system prompt of the CLI console, use the command:

```
Admin@nodename# set settings cli custom-prompt <new-custom-prompt>
```

To return the system prompt to its original state, use the command:

```
Admin@nodename# set settings cli custom-prompt default
```

UserGate General Settings

You configure UserGate server general settings at the **settings general** level. This is the command structure to configure one of the sections (<settings-module>):

```
Admin@nodename# set settings general <settings-module>
```

You can configure the following sections:

Parameter	Description
admin-console	Admin console settings (**settings general admin-console** level): • **timezone**: time zone for your location. Used in rule schedules and for the correct display of time and date in reports, logs, etc. • **language**: interface language: ◦ **ru**: Russian ◦ **en**: English • **uc-profile**: select the user certificate profile • **web-ssl-profile**: select an SSL profile to set up a secure channel to access the web console. For more details on SSL profiles, see Configuring SSL Profiles. • **response-pages-ssl-profile**: select an SSL profile to set up a secure channel to display web resource block pages and the Captive portal authorization page. For more details on SSL profiles, see Configuring SSL Profiles. • **api-session-lifetime**: admin session timeout in seconds.
server-time	

Parameter	Description
	Configure the exact time settings (**settings general server-time** level): • **ntp-enabled**: enable/disable the use of NTP servers: ◦ **on** ◦ **off** • **primary-ntp-server**: specify the primary ntp server. • **second-ntp-server**: specify a backup ntp server. • **time**: set server time (format: yyyy-mm-ddThh:mm:ss, e.g. 2022-02-15T12:00:00; UTC time zone).
modules	Configure UserGate modules (**settings general modules** level): • **proxy-port**: specify a non-standard port number for connecting to the built-in proxy server. • **auth-captive**: specify a service domain that UserGate uses to authorize users through the Captive portal. • **logout-captive**: specify a service domain that UserGate users use to end their session (logout). • **block-page-domain**: specify a service domain used to display the block page to users. • **ftp-enabled**: enable/disable the module that allows access to FTP server content from a user browser. • **ftp-domain**: specify a service domain to provide an FTP over HTTP connection to users. • **tunnel-inspection-zone**: select a tunnel inspection zone. You need to specify the following: ◦ **enabled**: enable/disable the zone ◦ **name**: specify the zone name • **snmp-engine-id**: configure SNMP Engine ID: ◦ **length** <**fixed** | **dynamic**>: fixed (8 bytes max; only for text type) or dynamic (27 bytes max.) ID length. ◦ **type** <**ip4** | **ip6** | **mac** | **text** | **octets**>: SNMP Engine ID format (IPv4, IPv6, MAC address, text, octets). ◦ **value**: the ID value. • **terminal-sever-agent**: configure the password for terminal server agents. • **lldp**: configure the use of Link Layer Discovery Protocol (LLDP), which allows the network equipment operating in a local network to notify devices about its existence, send its characteristics to them, and receive similar information from them. These settings are required: ◦ **transmit-delay**: how long the device will wait before sending advertisements to the neighbors

Parameter	Description
	after a change in the LLDP protocol's TLV parameter or the local system state (e.g., a changed hostname or management address). Specified in seconds and can take values from 1 to 3600. ◦ **transmit-hold**: the hold multiplier. The transmit delay multiplied by the transmit hold determines the time to live (TTL) for LLDP packets. Can take values from 1 to 100.
cache	Configure the proxy server cache (**settings general cache** level): • **caching-mode**: enable/disable caching. ◦ **on** ◦ **off** • **exclusions**: the list of URLs that will not be cached. To remove exclusions, use the following command: `Admin@nodename# delete settings general cache exclusions [ <URL> ]` • **max-cacheable-size**: maximum size of objects to be cached (in MB). • **ram-size**: RAM size allocated for caching (in MB).
log-analyzer	Log Analyzer module settings (**settings general log-analyzer** level): • **use-local-stat-server**: use the local Log Analyzer: ◦ **on** ◦ **off**
pcap	`Admin@nodename# set settings general pcap packet-capture-mode <parameter>` Configure packet capture (**settings general pcap** level): • **no-capture**: no capture. • **one-packet**: one packet. • **previous**: previous packets.

Parameter	Description
	• **previous-and-following**: previous and following packets. ◦ **previous-packets**: number of previous packets (from 4 to 30). ◦ **previous-packets**: number of following packets (from 2 to 15).
change-tracker	Configure change tracker (**settings general change-tracker** level): • **enabled**: enable/disable change tracker. ◦ **on** ◦ **off** • **event-tracker-types**: change types are set by an administrator. To delete a change type, use the following command: `Admin@nodename# delete settings general change-tracker event-tracker-types [ type1 ... ]`
management-center	`Admin@nodename# set settings general management-center <parameters>` Configure UserGate Management Center agent (**settings general management-center** level): • **enabled**: enable/disable the UserGate Management Center agent. ◦ **on** ◦ **off** • **mc-address**: UserGate Management Center server address. • **device-code**: unique device code to connect to the UserGate Management Center.
updates-schedule	Configure the schedule to download software and library updates (**settings general updates-schedule** level). To configure a schedule to update UserGate software, use the following command:

Parameter	Description
	`Admin@nodename# set settings general updates-schedule software schedule <schedule/disabled>` You can set up a single schedule to download library updates: `Admin@nodename# set settings general updates-schedule all-libraries schedule <schedule/disabled>` or an individual schedule for each item: `Admin@nodename# set settings general updates-schedule libraries [ lib-module ... ] schedule <schedule/disabled>` The time is set in the Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours". To view the update schedule, use the following command: `Admin@nodename# show settings general updates-schedule`
upstream-proxy	Configure HTTP redirection to an upstream proxy: • **enabled**: enable/disable traffic redirecting to an upstream proxy (on/off). • **mode**: the upstream proxy type (HTTP(S)/SOCKS5). • **ip**: the upstream proxy's IP address. • **port**: the upstream proxy's port. • **auth**: authentication with the upstream proxy (on/off).

Parameter	Description
	• **name**: the upstream proxy login name. • **password**: the upstream proxy password.

Configuring device management

Configuring diagnostics

At the **settings radmin** level, you can enable or disable remote access to the server for the UserGate technical support (**Radmin**). To enable/disable Radmin, use the following command:

```
Admin@nodename# set settings radmin enabled <on | off>
```

To view the Radmin state, use the following command:

```
 Admin@nodename# show settings radmin
```

The server diagnostics settings that the technical support team needs for troubleshooting are set at the **settings loglevel** level. You can use the following command to set the desired diagnostic details level (disabled; errors only; errors and warnings; errors, warnings, and additional information; maximum level of detail):

```
 Admin@nodename# set settings loglevel value <off | error | warning |
info | debug>
```

To view the status of the diagnostics detail level, use the following command:

```
Admin@nodename# show settings loglevel

value    : error
```

Configuring radmin emergency

To activate the remote assistant when a problem with the node's core software arises, the administrator can log in to the CLI using the root administrator account created when UserGate was initialized. Usually, this is the Admin account; however, it

is not always so. To log in, specify the name as Admin@emergency and use the root administrator password as the password. To enable/disable remote access to the server for technical support in such cases, use the following command:

```
Admin@nodename# set radmin-emergency enabled <on | off>
```

Parameter	Description
interface	The interface name.
ip-addr	Interface IP address and mask.
gateway-address	Gateway IP address.

Configuring server operations

To set an update channel, use the following command:

```
Admin@nodename# set settings device-mgmt updates-channel <stable |
beta>
```

To view any updates and the selected update channel, use the following command:

```
 Admin@nodename# show settings device-mgmt updates-channel
```

System backup management

A device backup is created at the **settings device-mgmt** level. To create a backup rule and upload files to external FTP/SSH servers, use the following command:

```
Admin@nodename# create settings device-mgmt settings-backup
<parameters>
```

The available parameters include:

Parameter	Description
enabled	Enable/disable the device backup rule.

Parameter	Description
name	The name of the backup rule.
description	A description of the backup rule.
type	Select a remote server to export files: • **ssh** • **ftp**
address	Remote server IP address.
port	Port:
login	Remote server login name.
password	Password for the login name.
path	Directory path to upload the files to.
schedule	The backup file export schedule. The time is set in the Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

To edit an existing UserGate device backup rule, use the following command:

```
Admin@nodename# set settings device-mgmt settings-backup <rule-name>
```

You can use the same set of parameters as when creating rules.

To delete a backup rule:

```
Admin@nodename# delete settings device-mgmt settings-backup <rule-
name>
```

To display a backup rule:

```
Admin@nodename# show settings device-mgmt settings-backup <rule-name>
```

In the rule edit, delete, or display commands, <filter> can include the parameters specified in an existing rule in addition to the rule name (this can be helpful if there are multiple rules with the same name). Parameters used to identify an export rule are similar to those of the **set** command.

Settings Export

You create and configure export settings rules at the **settings device-mgmt settings-export** level.

To create an export settings rule, use the following command:

```
Admin@nodename# create settings device-mgmt settings-export
( <parameters> )
```

Available parameters:

Parameter	Description
enabled	Enable/disable an export settings rule for the UserGate server.
name	Export rule name.
description	Export rule description.
type	Select a remote server to export settings: • **ssh** • **ftp**
address	Remote server IP address.
port	Port:
login	Remote server login name.

Parameter	Description
password	Password for the login name.
path	Directory path to upload the settings to.
schedule	Schedule for settings export. The time is set in the Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

To update an existing rule to export UserGate server settings, use the following command:

```
Admin@nodename# set settings device-mgmt settings-export <rule-name>
```

You can use the same set of parameters as when creating rules.

To delete a rule to export settings, use the following command:

```
Admin@nodename# delete settings device-mgmt settings-export <rule-
name>
```

To display a rule to export settings, use the following command:

```
Admin@nodename# show settings device-mgmt settings-export <rule-name>
```

For update, delete or display rule commands, you can set <filter> not only to the rule name, but also to the parameters specified in an existing rule (this may be helpful if

there is more than one rule with the same name). Parameters used to identify an export rule are similar to those of the **set** command.

Settings for protecting configuration data from changes

To configure settings for protecting product configuration data (settings) from being changed, use the following command:

```
Admin@nodename# set settings change-control config <off | log | block>
```

Configuration data integrity is checked every few minutes after UserGate boots.

- **log**: enable configuration change tracking. If any changes are detected, UserGate records this information in the event log. A password is required which will be used to change the tracking mode.
- **off**: disable configuration change tracking. Requires the password that was set when enabling the configuration change tracking.
- **block**: activate configuration change tracking. A password is required which will be used to change the tracking mode. If any changes are detected, UserGate records this information in the event log and creates a firewall blocking rule that denies any transit traffic through UserGate.

Before enabling configuration data protection, the administrator configures the product according to the organization's requirements and then "freezes" the settings (**log** or **block** mode). Any setting change through the web interface, CLI, or other means will result in logging and/or blocking of transit traffic, depending on the selected mode.

To view the current configuration data protection mode, use the following command:

```
Admin@nodename# show settings change-control config
```

Protect executable files from changes

To configure settings to protect product executable code from potential unauthorized modification, use the following command:

```
Admin@nodename# set settings change-control code <off | log | block>
```

Executable code integrity is checked every few minutes after UserGate boots.

- **log**: enable the tracking of unauthorized changes in executable code. If any changes are detected, UserGate records this information in the event log. A password is required which will be used to change the tracking mode.
- **off**: disable the tracking of unauthorized changes in executable code. Requires the password that was set when enabling the executable code change tracking.
- **block**: enable the tracking of unauthorized changes in executable code. A password is required which will be used to change the tracking mode. If any changes are detected, UserGate records this information in the event log and creates a firewall blocking rule that denies any transit traffic through UserGate. To disable an existing firewall rule you need to disable tracking of unauthorized changes.

To view the current executable file protection mode, use the following command:

```
Admin@nodename# show settings change-control code
```

Configuring Accelerated Network Traffic Processing Mode

To enable/disable the accelerated traffic processing mode, use the command:

```
Admin@nodename# set settings fastpath enabled <on/off>
```

To view the settings for the accelerated traffic processing mode, use the command:

```
Admin@nodename# show settings fastpath
```

Cluster Settings

Configuration cluster settings

This section is located at the **settings device-mgmt configuration-cluster** level.

To update an existing cluster mode, use the following command:

```
Admin@nodename# set settings device-mgmt configuration-cluster <node-
name>
```

Available parameters:

Parameter	Description
name	Change the cluster node name.
description	Update the cluster node description.
ip	Set the IP address of the interface included in the zone allocated to the cluster.

To delete and display cluster node settings, use the following commands:

```
Admin@nodename# delete settings device-mgmt configuration-cluster
<node-name>
...
Admin@nodename# show settings device-mgmt configuration-cluster <node-
name>
```

To generate a secret code for adding a new node to the configuration cluster, use the following command:

```
Admin@nodename# execute configurate-cluster generate-secret-key
```

Settings for high availability clusters

You apply settings to HA clusters at the **settings device-mgmt ha-cluster** level.

To create an HA cluster, use the following command:

```
Admin@nodename# create settings device-mgmt ha-clusters
```

Provide the following parameters:

Parameter	Description
enabled	Enable/disable the HA cluster: • **on** • **off**
name	HA cluster name.
description	HA cluster description.
mode	Select cluster operation mode: • **active-passive**: Active-Passive mode (one server operates as the master node that processes traffic while the remaining servers act as backup). • **active-active**: Active-Active mode (one server operates as the master node that distributes traffic to all other nodes in the cluster).
session-sync	Configure user session synchronization in the cluster: • **off**: disable user session synchronization • **on**: enable user session synchronization • **ha-cluster-id:** ◦ <num>: HA cluster multicast ID (can take values of 0 to 8). User session synchronization (except for sessions that use a proxy server, such as HTTP/S traffic) is enabled automatically.
virtual-router-id	Virtual Router ID (VRID).
nodes	Select configuration cluster nodes to combine them into an HA cluster.
virtual-ips	Set the virtual IP address for the cluster and select an interface for each node (the VRRP service should be enabled in the selected interface zone; for more details on how to configure zones using the CLI, see the Zones section). To add a virtual IP address to the cluster, use the following command: `Admin@nodename# create settings device-mgmt ha-cluster virtual-ips <virtual-ips-filter> <virtual-ip-info>`

Parameter	Description
	Available parameters for <virtual-ips-filter>: • **new**: create a virtual IP address for the specific cluster. • <ip>: change data for the selected virtual address. Available parameters for <virtual-ip-info>: • **ip**: set an IP address for the HA cluster (format: IP/mask). • **ha-interfaces**: set interfaces for the cluster nodes (format: node-name/interface).
session-sync-all	Enable/disable synchronizing all user sessions, including UDP/ICMP sessions. If this is disabled and **session-sync** enabled, only TCP sessions will be synchronized.
excluded-sync-ips	Specify the IP for which synchronization is disabled for all user sessions.

Example cluster creation command:

```
Admin@nodename# create settings device-mgmt ha-clusters nodes
[ node_1 ] name "Test HA cluster" description "Test HA cluster
description" mode active-passive enabled on virtual-ips new ha-
interfaces [ node_1/port3 ] ip 192.168.1.5/24
```

To edit the cluster settings, use the following command:

```
Admin@nodename# set settings device-mgmt ha-cluster <cluster-name>
```

The following parameters are available:

Parameter	Description
enabled	Enable/disable the HA cluster: • **on** • **off**
name	HA cluster name.
description	HA cluster description.

Parameter	Description
mode	Select cluster operation mode: • **active-passive**: Active-Passive mode (one server operates as the master node that processes traffic while the remaining servers act as backup). • **active-active**: Active-Active mode (one server operates as the master node that distributes traffic to all other nodes in the cluster).
master-node	Assign the master node in the HA cluster.
session-sync	Configure session synchronization in the cluster: • **off**: disable user session synchronization • **on**: enable user session synchronization • **ha-cluster-id:** ◦ <num>: HA cluster multicast ID (can take values of 0 to 8). User session synchronization (except for sessions that use a proxy server, such as HTTP/S traffic) is enabled automatically.
virtual-router-id	Virtual Router ID (VRID).
nodes	Select configuration cluster nodes to combine them into an HA cluster.
virtual-ips	Set the virtual IP address for the cluster and select an interface for each node (the VRRP service should be enabled in the selected interface zone; for more details on how to configure zones using the CLI, see the Zones section). To add a virtual IP address to the cluster, use the following command: `Admin@nodename# create settings device-mgmt ha-cluster virtual-ips <virtual-ips-filter> <virtual-ip-info>` Available parameters for <virtual-ips-filter>: • **new**: create a virtual IP address for the specific cluster. • <ip>: change data for the selected virtual address. Available parameters for <virtual-ip-info>: • **ip**: set an IP address for the HA cluster (format: IP/mask).

Parameter	Description
	• **ha-interfaces**: set interfaces for the cluster nodes (format: node-name/interface).
session-sync-all	Enable/disable synchronizing all user sessions, including UDP/ICMP sessions. If this is disabled and **session-sync** enabled, only TCP sessions will be synchronized.
excluded-sync-ips	Specify the IP for which synchronization is disabled for all user sessions.

Example commands for editing the cluster settings:

```
Admin@nodename# set settings device-mgmt ha-clusters "Test HA cluster"
nodes [ node_1 node_2 ] virtual-ips 192.168.1.5/24 ha-interfaces
[ node_1/port3 node_2/port3 ]
...
Admin@nodename# set settings device-mgmt ha-clusters "Test HA cluster"
master-node utmcore@iononsteswer
```

To delete a cluster, use the following command:

```
Admin@nodename# delete settings device-mgmt ha-clusters <cluster-name>
```

You can also delete individual parameters:

- **nodes**
- **virtual-ips**

To display information about all HA clusters, use the following command:

```
Admin@nodename# show settings device-mgmt ha-cluster
```

To display information about a specific HA cluster, use the following command:

```
Admin@nodename# show settings device-mgmt ha-cluster <cluster-name>
```

Configuring the UserGate DCFW console access management

This section is configured at the **settings administrators** level. This section describes how to configure account security settings, administrators, and their profiles.

General access settings

In this section, you can configure additional security options for administrator accounts. This is configured at the **settings administrators general** level.

To change the parameters, use the following command:

```
Admin@nodename# set settings administrators general
```

The following parameters are available:

Parameter	Description
password	Change the current administrator password.
unblock	Unblock an administrator.
strong-password	Use a strong password: • **on** • **off**
num-auth-attempts	Maximum number of incorrect authentication attempts.
block-time	Time to block an account if the maximum number of authentication attempts is reached by the administrator (in seconds, max value is 3600 seconds).
min-length	Minimum password length (max value is 100 characters).
min-uppercase	Minimum number of uppercase characters (max value is 100 characters).
min-lowercase	Minimum number of lowercase characters (max value is 100 characters).
min-digits	Minimum number of digits (max value is 100 characters).

Parameter	Description
spec-characters	Minimum number of special characters (max value is 100 characters).
char-repetition	Maximum single character repetition block length (max value is 100 characters).

Examples of editing account parameters:

```
Admin@nodename# set settings administrators general block-time 400
```

To view the current security settings for administrator accounts, use the following command:

```
Admin@nodename# show settings administrators general

strong-password      : off
block-time           : 400
min-length           : 7
min-uppercase        : 1
min-lowercase        : 1
min-digits           : 1
spec-characters      : 1
char-repetition      : 2
num-auth-attempts    : 10
```

Configuring administrator accounts

You configure administrator accounts at the **settings administrators administrators** level.

To create an administrator account, use the following command:

```
Admin@nodename# create settings administrators administrators
```

Specify the administrator account type (local, LDAP user, LDAP group, with auth profile) and the respective parameters:

Parameter	Description
local	Add a local administrator: • **enabled**: enable/disable an administrator account: ◦ **on** ◦ **off** • **login**: administrator login name. • **description**: administrator account description. • **admin-profile**: administrator profile. For more details about creating administrator profiles, see below. • **password**: administrator password.
ldap-user	Add a user from the existing domain (you need to have the LDAP connector configured correctly; for more details, see the Configuring LDAP Connectors section): • **enabled**: enable/disable an administrator account: ◦ **on** ◦ **off** • **login**: the administrator's login name in the **domain\user** format. When providing this parameter, use the following command structure: • **connector**: the name of a previously configured LDAP connector. • **description**: administrator account description. • **admin-profile**: administrator profile. For more details about creating administrator profiles, see below. `Admin@nodename# create settings administrators administrators ldap-user admin-profile "test profile 1" connector "LDAP connector" description "Admin as domain user" login testd.local\user1 enabled on`
ldap-group	Add a user group from the existing domain (you need to have the LDAP connector configured correctly; for more details, see the Configuring LDAP Connectors section): • **enabled**: enable/disable an administrator account: ◦ **on** ◦ **off** • **login**: administrator login name • **connector**: the name of the used LDAP connector.

Parameter	Description
	• **description**: administrator account description. • **admin-profile**: administrator profile. For more details about creating administrator profiles, see below. `Admin@nodename# create settings administrators administrators ldap-group admin-profile "test profile 1" connector "LDAP connector" description "Domain admin group" login testd.local\users enabled on`
admin-auth-profile	Add an administrator with an auth profile (you need to have the auth servers configured correctly; for more details, see the Configuring Authentication Servers section): • **enabled**: enable/disable an administrator account: ◦ **on** ◦ **off** • **login**: administrator login name. • **description**: administrator account description. • **admin-profile**: administrator profile. For more details about creating administrator profiles, see below. • **auth-profile**: select an auth profile from those created earlier; for more details about auth profiles, see the section Configuring Authentication Profiles.

To edit the profile parameters, use the following command:

```
Admin@nodename# set settings administrators administrators <admin-type> <admin-login>
```

The command's parameters are similar to those used for administrator profile creation.

To display information about all administrator accounts, use the following command:

```
Admin@nodename# show settings administrators administrators
```

To display information about an individual administrator account, use the following command:

```
Admin@nodename# show settings administrators administrators <admin-
type> <admin-login>
```

Example of the command execution:

```
Admin@nodename# show settings administrators administrators ldap-user
testd.local\user1

login            : testd.local\user1
enabled          : on
type             : ldap_user
locked           : off
admin-profile    : test profile 1
```

To delete an account, use the following command:

```
Admin@nodename# delete settings administrators administrators <admin-
type> <admin-login>
```

Example of the command:

```
Admin@nodename# delete settings administrators administrators ldap-user
testd.local\user1
```

Configuring Permissions for Administrator Profiles

The permissions of administrator profiles are configured at the **settings administrators profiles** level.

To create an administrator profile, use the following command:

```
Admin@nodename# create settings administrators profiles
```

Provide the following parameters:

Parameter	Description
name	Administrator profile name.
description	Administrator profile description.
api-permissions	API permissions: • **no-access**: no access • **read**: read-only • **write**: read and write You can assign rights to all or individual objects: `Admin@nodename# create settings administrators profiles ... api-permissions <permission> all` or `Admin@nodename# create settings administrators profiles ... api-permissions <permission> [ object ... ]`
webui-permissions	UserGate interface permissions: • **no-access**: no access • **read**: read-only • **write**: read and write You can assign rights to all or individual objects: `Admin@nodename# create settings administrators profiles ... webui-permissions <permission> all` or `Admin@nodename# create settings administrators profiles ... webui-permissions <permission> [ object ... ]`
cli-permissions	Command line interface permissions: • **no-access**: no access • **read**: read-only

Parameter	Description
	• **write**: read and write You can assign rights to all or individual objects: `Admin@nodename# create settings administrators profiles ... cli-permissions <permission> all` or `Admin@nodename# create settings administrators profiles ... cli-permissions <permission> [ object ... ]`

To edit the profile, use the following command:

```
Admin@nodename# set settings administrators profiles <profile-name> <parameter>
```

The command's parameters are similar to those used for administrator profile creation.

To view information about all administrator profiles, use the following command:

```
Admin@nodename# show settings administrators profiles
```

To display information about a specific profile, use the following command:

```
Admin@nodename# show settings administrators profiles <profile-name>
```

To delete an administrator profile, use the following command:

```
Admin@nodename# delete settings administrators profiles <profile-name>
```

Managing Administrator Sessions

The following commands allow you to view the active sessions of administrators who have been authorized in the web console or CLI and close the sessions (this is done at the **settings administrators admin-sessions** level).

To view administrator sessions for the current UserGate node, use the following command. You can view an individual administrator's session; to do so, browse the IP address list and select the address used to authenticate the administrator.

```
Admin@nodename# show settings administrators admin-sessions
```

To display sessions, you can use a filter:

- **ip**: IP address from which the administrator was authorized.
- **source**: where authorization was made: CLI (**cli**), web console (**web**) or SSH connection (**ssh**).
- **admin-login**: administrator name.
- **node**: UserGate cluster node.

```
Admin@nodename# show settings administrators admin-sessions ( node
<node-name> ip <session-ip> source <cli | web | ssh> admin-login
<administrator-login> )
```

To close an administrator session, use the following command. Select the IP address from which the administrator was authorized, from the list.

```
Admin@nodename# execute termination admin-sessions <IP-address/
connection type>
```

Example of the command execution:

```
Admin@nodename# show settings administrators admin-sessions

admin-login           : Admin
source                : ssh
```

```
session_start_date    : 2023-08-10T11:33:47Z
ip                    : 127.0.0.1
node                  : utmcore@dineanoulwer

admin-login           : Admin
source                : web
session_start_date    : 2023-08-10T11:33:10Z
ip                    : 10.0.2.2
node                  : utmcore@dineanoulwer

Admin@nodename# execute termination admin-sessions 10.0.2.2/web

Admin@nodename# show settings administrators admin-sessions

admin-login           : Admin
source                : ssh
session_start_date    : 2023-08-10T11:33:47Z
ip                    : 127.0.0.1
node                  : utmcore@dineanoulwer
```

When closing administrator sessions, you can use a filter (<filter>). Enabled filtering options are the same as those for the **show** command.

```
 Admin@nodename# execute termination admin-sessions ( node <node-name>
ip <session-ip> source <cli | web | ssh> admin-login <administrator-
login> )
```

Configuring Certificates

The **Certificates** section is located at the **settings certificates** level.

To import certificates, use the following command:

```
Admin@nodename# import settings certificates
```

Parameters:

Parameter	Description
name	Certificate name that will be listed.
description	Certificate description.
certificate-data	Certificate in PEM format.
certificate-chain	Certificate's chain in PEM format.
private-key	Private key in PEM format.
passphrase	Passphrase for the private key or PKCS12 container (optional value).
user	Local user to which the user certificate will be assigned.
ldap-user	LDAP connector user to which the user certificate will be assigned. • **user**: user name in domain\user format. • **connector**: select an LDAP server.
role	Certificate type: • **web-cert-chain**: web console certificate's chain. • **ssl-intermediate**: an intermediate certificate in the certification authority chain. • **ssl-root**: a root certificate in the certification authority chain. • **user**: user certificate that can be used to authenticate users when they access published resources using reverse proxy rules. • **captive-portal**: certificate used to create a secure HTTPS connection for users to the Captive portal authentication page, to display the block page, the Captive portal Logout page, and to operate an FTP proxy. • **web-ssl**: certificate used to create a secure HTTPS administrator connection to the UserGate web console. • **saml**: certificate the SAML client will use. • **none**.

To export certificates, the entire certificate's chain or CSR, use the following command:

```
Admin@nodename# export settings certificates <certificate-name>
Admin@nodename# export settings certificates <certificate-name> with-
chain on
```

To create a certificate and CSR, use the following command:

```
Admin@nodename# create settings certificates type <certificate | csr>
```

Provide the following parameters:

Parameter	Description
name	Certificate name.
description	Certificate description.
country	Country where the certificate is being issued.
state	Region/state where the certificate is being issued.
locality	Locality name where the certificate is being issued.
organization	Organization name for which the certificate is being issued.
common-name	Certificate name. To ensure compatibility with the majority of browsers, we recommend using only Latin characters.
email	Company email.

To manage a certificate, use the following command:

```
Admin@nodename# set settings certificates <certificate-name>
```

Available parameters:

Parameter	Description
name	Certificate name.
description	Certificate description.
role	

Parameter	Description
	Certificate type: • **web-cert-chain**: web console certificate's chain. • **ssl-intermediate**: an intermediate certificate in the certification authority chain. • **ssl-root**: a root certificate in the certification authority chain. • **user**: user certificate that can be used to authenticate users when they access published resources using reverse proxy rules. • **captive-portal**: certificate used to create a secure HTTPS connection for users to the Captive portal authentication page, to display the block page, the Captive portal Logout page, and to operate an FTP proxy. • **web-ssl**: certificate used to create a secure HTTPS administrator connection to the UserGate web console. • **saml**: certificate the SAML client will use. • **none**.
user	Local user to which the user certificate will be assigned.
ldap-user	LDAP connector user to which the user certificate will be assigned. • **user**: user name in domain\user format. • **connector**: select an LDAP server.
certificate-data	Certificate in PEM format.
certificate-chain	Certificate's chain in PEM format.

To delete a certificate, use the following command:

```
Admin@nodename# delete settings certificates <certificate-name>
```

To view information about all or individual certificates, use the following command:

```
Admin@nodename# show settings certificates
Admin@nodename# show settings certificates <certificate-name>
```

To delete a certificate from the cache, use the following command:

```
Admin@nodename# delete settings certificates-cache <common-name>
```

Settings for Device Parameters

You change device parameters at the **settings device** level. To change a device parameter, use the following command (the <setting-name> is the parameter name):

```
Admin@nodename# set settings device <setting-name>
```

Available parameters:

Parameter	Description
l7	Enable/disable L7 module load: • **on** • **off** By default, the module is loaded. **Important!** If you change this parameter, you need to reboot your UserGate device.
sip	Enable/disable SIP module load. The module needs to be enabled to map the signaling and data connection when NAT is used: • **on** • **off** By default, the module is unloaded. **Important!** After enabling, for the module to work correctly, you must reload the firewall rules table (the **Force changes** button in the **Network Policies → Firewall** section).
h323	Enable/disable h323 module load. The module needs to be enabled to map the signaling and data connection when NAT is used: • **on** • **off** By default, the module is unloaded.
idps	

Parameter	Description
	Enable/disable IDPS module load: • **on** • **off** By default, the module is loaded. **Important!** If you change this parameter, you need to reboot your UserGate device.
sunrpc	Enable/disable SunRPC module load: • **on** • **off** By default, the module is unloaded.
ftp-alg	Enable/disable FTP module load. The module needs to be enabled to map the signaling and data connection when NAT is used: • **on** • **off** **Important!** The module must be enabled for passive FTP mode. By default, the module is unloaded.
auth-type	Use the IPsec Authentication Header signature for VRRP service packets in an HA cluster: • **ah**: enable the signature • **pass**: disable signature checking.
fw-drop-invalid	Enable/disable blocking of packets with an invalid parameter set in the header fields: • **on** • **off** The default setting is off. Enabling this option significantly reduces the firewall performance, so we recommend to leave this setting disabled.
fw-established	Enable/disable creation of a single common firewall rule for return packets: • **on** • **off**

Parameter	Description
	The default setting is off.
bypass-optimization	Enable/disable SSL inspection optimization: • **on** • **off** The default setting is off.

To view the current settings, use the following command:

```
Admin@nodename# show settings device
```

Configuring Device Monitoring Settings

Configuring device monitoring parameters in the CLI interface is done in configuration mode at the **monitoring** level. Commands at this level allow you to manage the configuration of SNMP device parameters, SNMP monitoring rules, security profiles for authenticating SNMP managers, and notification rules. Read more about monitoring and notification rules in the Notifications section.

Configuring SNMP Device Parameters

To configure the SNMP device parameters, use commands at the **monitoring smnp-parameter** level:

```
Admin@nodename# edit monitoring snmp-parameter <parameters>
```

You can edit the following parameters:

Name	Description
agent-name	Name of the system which is used by SNMP control subsystem.
location	Information on physical location of the SNMP agent.
description	Description of the system.
Engine ID	

Name	Description
	Each UserGate device has a unique SNMPv3 Engine ID. By default, the Engine ID is generated from the UserGate node name. When editing the Engine ID, you are required to specify its length (**length**), type, and value. The length can be defined as fixed (max. 8 bytes) or dynamic (max. 27 bytes). A fixed ID length is only applicable to the **text** type. The Engine ID can be generated in these formats: • **ip4**: IPv4 • **ipv6**: IPv6 • **mac**: MAC address • **text**: text • **octets**: octets

Read more about the SNMP parameters of the UserGate device in the SNMP section.

Configuring SNMP Monitoring Rules

To configure device monitoring rules via SNMP, commands are used at the **monitoring snmp** level:

```
Admin@nodename# edit monitoring snmp <parameters>
```

You can edit the following parameters:

Name	Description
name	The name of the rule.
enabled	Enable/disable a rule
community	SNMP community — the string for UserGate server identification and SNMP server identification for SNMP v2c. Use only Latin letters and numbers.
context	Optional parameter that defines the SNMP context. Use only Latin letters and numbers. Some devices may have multiple copies of the entire MIB subtree. For example, several virtual routers can be created on the device. Each such virtual router will have a complete MIB subtree. In this case, each virtual router can be specified as a context on the SNMP server. The context is identified by name. When the client makes a request, the context name can be

Name	Description
	specified. If the context name is not specified, the default context will be requested.
version	Specify the version of the SNMP protocol used in the rule. Available options: SNMP v2c and SNMP v3.
query	When enabled, allows receiving and processing of SNMP requests from the SNMP manager.
trap	When enabled, allows sending of SNMP traps to the server configured to receive notifications.
trap-host	Server IP address for traps. This setting is required only if you need to send traps to the notification server.
trap-port	The port on which the server listens for notifications. Usually, it is UDP port 162. This setting is required only if you need to send traps to the notification server.
security-profile	For SNMP v3 only. For more details, see the SNMP Security Profiles section.
events	Selecting the types of parameters available for monitoring by rule.

To use the SNMP manager with UserGate DCFW, you need to enable the **SNMP** service in the access control settings in the zone properties of an interface which will be connected to using the SNMP. For more information about setting up zones in the CLI, see the Network Settings section.

Configuring SNMP Security Profiles

To configure security profiles to authenticate SNMP managers, use commands at the **monitoring smnp-security-profile** level:

```
Admin@nodename# edit monitoring snmp-security-profile <parameters>
```

You can edit the following parameters:

Name	Description
name	SNMP security profile name
description	SNMP security profile description

Name	Description
username	User name to authenticate the SNMP manager.
auth-type	Select an authentication mode for the SNMP manager. The available options are: • **none**: no authentication, no encryption • **no-encrypt** : authentication, no encryption • **encrypt**: authentication, encryption The authPriv mode is considered the most secure.
auth-alg	The algorithm used for authentication. Possible to use: • **sha** • **md5** • **sha224** • **sha256** • **sha384** • **sha512**
auth-password	The password used for authentication.
encrypt-alg	The algorithm used for encryption. DES or AES can be used.
encrypt-password	The password used for encryption.

Configuring Notification Rules

To configure alert rules, use commands at the **monitoring alert-rules** level:

```
Admin@nodename# edit monitoring alert-rules <parameters>
```

You can edit the following parameters:

Name	Description
enabled	Enables/disables the rule.
name	The name of the rule.
description	A description of the rule.

Name	Description
notification-profile	A previously created notification profile.
sender	From whom the notifications will come.
subject	Notification subject.
timeout	The timeout during which the server will not send a message when this rule is triggered again. This setting prevents a flood of messages when an alert rule is triggered frequently.
events	Events for which you want to receive alerts.
phones	For SMPP profiles, The phone groups to which SMS notifications will be sent.
emails	For SMTP profiles. The groups of email addresses to which email notifications will be sent.

These are the options for packet capture

The Packet capture allows you to record the traffic that meets the specified conditions to a PCAP file for further analysis using third-party tools, such as wireshark. This may be necessary while diagnosing network problems.

Pcap filters determine the conditions under which traffic will be recorded. You can use a source address, a source port, a destination address, a destination port, or an IPv4 protocol as conditions.

To configure pcap filters, use commands at the **monitoring pcap-filter** level:

```
Admin@nodename# edit monitoring pcap-filter <parameters>
```

The pcap rules specify the UserGate interfaces on which traffic must be recorded, the filters created earlier, the name and size of the file in which the intercepted traffic is recorded.

To configure pcap rules, use commands at the **monitoring pcap-rule** level:

```
Admin@nodename# edit monitoring pcap-rule <parameters>
```

Configuring Client Certificate Profiles

The **Client certificate profiles** section is located at the **settings certificate-profiles** level.

To create a client certificate profile, use the following command:

```
Admin@nodename# create settings certificate-profiles <parameters>
```

The following parameters can then be used:

Parameter	Description
name	The name of the client certificate profile.
description	Profile description.
username-field	Select the field in the certificate that determines the username used for authentication: • **common**: the domain name or hostname in the Subject field for which the certificate is intended. • **email**: the parameter with the email prefix in the SAN (Subject Alternative Name) extension is used to determine the username. • **principal**: the Universal Principal Name (UPN) parameter contained in the otherName field in the SAN extension is used to determine the username. If multiple UPNs or email addresses are specified in the SAN extension fields of a certificate, the first one specified in the certificate is used.
certificates	The CA certificates assigned to the profile.
crl	Certificate revocation lists (CRLs) contain certificates that have been revoked and can no longer be used. This list includes certificates that have expired or been compromised. Certificate revocation status check method: • **off**: do not check any certificates. • **on**: check all certificates in the chain and require that they are all valid. • **peer**: check only the client certificate. • **best-effort**: if the CRL could not be verified for some reason, then the certificate is considered valid (however,

Parameter	Description
	it is still checked and may return the invalid status if the certificate is on the revocation list).
receive-timeout	The time interval after which DCFW stops waiting for the response from the certificate revocation list service.

To view previously created client certificate profiles, use the following commands:

```
Admin@nodename# show settings certificate-profiles
Admin@nodename# show settings certificate-profiles <certificate-profile-name>
```

To edit the previously created profile, use the following command:

```
Admin@nodename# set settings certificate-profiles <certificate-profile-name> <parameters>
```

The editable profile parameters are the same as those for creating a profile discussed earlier.

To delete the previously created profile, use the following command:

```
Admin@nodename# delete settings certificate-profiles <certificate-profile-name>
```

NETWORK CONFIGURATION

Zones

This section is located at the **network zone** level. To create a new zone, use the following command:

```
Admin@nodename# create network zone
```

Provide the following zone parameters:

Parameter	Description
name	Zone name.
description	Zone description.
dos-protection-syn	Protect the zone against network flooding for TCP protocol (SYN-flood): • **enabled**: enable/disable the protection. ◦ **on** ◦ **off** • **aggregate**: ◦ **on**: count all packets incoming to the zone's interfaces ◦ **off**: count packets for each IP address separately. • **alert-threshold**: alert threshold; if the number of requests exceeds this value, the event is recorded in the system log. • **drop-threshold**: packet drop threshold; if the number of requests exceeds this value, UserGate drops packets and records this event in the system log. • **excluded-ips**: list of IP addresses of servers that should be excluded from protection.
dos-protection-udp	Protect the zone against network flooding for UDP protocol: • **enabled**: enable/disable the protection. ◦ **on** ◦ **off** • **aggregate**: ◦ **on**: count all packets incoming to the zone's interfaces ◦ **off**: count packets for each IP address separately. • **alert-threshold**: alert threshold; if the number of requests exceeds this value, the event is recorded in the system log. • **drop-threshold**: packet drop threshold; if the number of requests exceeds this value, UserGate drops packets and records this event in the system log. • **excluded-ips**: list of IP addresses of servers that should be excluded from protection.

Parameter	Description
dos-protection-icmp	Protect the zone against network flooding for ICMP protocol: • **enabled**: enable/disable the protection. ◦ **on** ◦ **off** • **aggregate**: ◦ **on**: count all packets incoming to the zone's interfaces ◦ **off**: count packets for each IP address separately. • **alert-threshold**: alert threshold; if the number of requests exceeds this value, the event is recorded in the system log. • **drop-threshold**: packet drop threshold; if the number of requests exceeds this value, UserGate drops packets and records this event in the system log. • **excluded-ips**: list of IP addresses of servers that should be excluded from protection.
enabled-services	Zone access control settings: • **"Any ICMP"**: allow use of the ping command to a UserGate address. • **SNMP**: provides SNMP access to UserGate (UDP 161). • **response-pages**: permission to display Captive portal auth and block pages (TCP 80, 443, 8002). • **rpc**: control XML-RPC: enables API control of the product (TCP 4040). • **ha**: service required to combine multiple UserGate nodes into a cluster (TCP 4369, TCP 9000-9100). • **VRRP**: required for combining several UserGate nodes into a HA cluster (IP protocol 112). • **"Admin Console"**: access to the management web console (TCP 8001). • **"Authorization agent"**: server access required for Windows authorization agents and terminal servers (UDP 1813). • "CLI over SSH": access to server to manage it via CLI, port TCP 2200. • **VPN**: provides server access for connecting L2TP VPN clients (UDP 500, 4500). • **L7 DNS**: DNS traffic detection at the application level. • **L7 NTP**: NTP traffic detection at the application level. • **"SAML SERVER"**: select an SAML server in the list of zone services and general UserGate settings.

Parameter	Description
	• **Log Analyzer**: the Log Analyzer service. Enable this if you plan to use this UserGate server as a Log analyzer (TCP 2023 and 9713). • **"Dynamic routing OSPF"**: OSPF dynamic routing service. • "Dynamic routing BGP": BGP dynamic routing service. • **"SNMP Proxy"**: service used to build a distributed monitoring system (used to balance load and organize monitoring of a distributed network infrastructure). • **Multicast**: multicast service. • **NTP**: access to the accurate time service running on the UserGate server. • **"Dynamic routing RIP"**: RIP dynamic routing service. • **UserID agent**: a transparent authentication service. Active Directory log and Syslog are used as the authentication data source for that purpose. • **BFD**: the Bidirectional Forwarding Detection service for quick network fault detection.
service-addresses	Allowed IP addresses for services: • **service**: select services (the list corresponds to **enabled-services**). • **allowed-addresses**: the allowed IP addresses. The options are: ◦ **geoip**: a GeoIP code ◦ **ip-list**: an IP address list previously configured in the item library.
antispoof-enabled	Enable/disable IP spoofing protection: • **on** • **off**
antispoof-negate	Enumerated options: • **on** • **off** If **antispoof-negate on** is enabled, the interfaces in that zone will not receive packets from the source addresses specified in the value **ip-spoofing-networks**. In this case packets with specified source IP addresses will be discarded.
sessions-limit-enabled	

Parameter	Description
	Enable the limit on the number of concurrent sessions from a single IP address: • **on** • **off**
sessions-limit-exclusions	Add a list of IP addresses to which the concurrent session limit will not apply.
sessions-limit-threshold	The maximum allowed number of sessions originating from a single IP address.
geoip	GeoIP codes that are used in IP spoofing protection.
ip-list	List of IP addresses that are used in IP spoofing protection.

Example command to create a zone:

```
Admin@nodename# create network zone name Test_zone description
"Test_zone description" antispoof-enable on enabled-services [ "Any
ICMP" DNS ] dos-protection-icmp enabled on
```

To edit zone parameters, use the following command:

```
Admin@nodename# set network zone <zone-name>
```

To edit zone parameters, use the following command:

```
Admin@nodename# set network zone Test_zone dos-protection-syn enabled
on
```

To delete a zone or its parameters, use the following command:

```
Admin@nodename# delete network zone <zone-name>
```

You can delete the following parameters:

Parameter	Description
dos-protection-syn	Protect the zone against network flooding for TCP protocol (SYN-flood): • **excluded-ips**: list of IP addresses of servers that should be excluded from protection.
dos-protection-udp	Protect the zone against network flooding for UDP protocol: • **excluded-ips**: list of IP addresses of servers that should be excluded from protection.
dos-protection-icmp	Protect the zone against network flooding for ICMP protocol: • **excluded-ips**: list of IP addresses of servers that should be excluded from protection.
enabled-services	The previously configured zone access control settings
geoip	GeoIP codes that are used in IP spoofing protection.
ip-list	List of IP addresses that are used in IP spoofing protection.

To preview zone settings, use the following command:

```
Admin@nodename# show network zone <zone-name>
```

Interfaces

An ordered list of network interface names with the associated physical addresses can be displayed using this command (available both in the Diagnostics and monitoring and Configuration modes):

```
Admin@nodename> show network interface-mapping

Admin@nodename# show network interface-mapping
```

The interfaces are ordered b port number on the PCI bus.

To delete the list, use the following commands (available both in the Diagnostics and monitoring and Configuration modes):

```
Admin@nodename> clear network interface-mapping

Admin@nodename# delete network interface-mapping
```

After the UserGate server reboots, the list will update and become available for display. This operation needs to be performed after adding network ports to a configured UserGate appliance.

Discussed next is interface configuration, which is done at the **network interface** level.

Adapter settings

Network adapters are configured at the **network interface adapter** level.

You cannot create a network adapter. To update an existing network adapter, use the command:

```
Admin@nodename# set network interface adapter <adapter_name>
```

Provide the following network adapter parameters:

Parameter	Description
enabled	Enable/disable a network interface: • **on** • **off**
description	Network interface description.
alias	The interface's alias.

Parameter	Description
iface-type	Interface type: • **l3**: interface works in Layer 3 mode (you can assign an IP address and use it in firewall, content filtering, and other rules; this is the standard interface operation mode). • **mirror**: interface works in Mirror mode (it can receive traffic from the network equipment SPAN port to analyze it).
iface-mode	IP address assignment mode: • **dhcp**: obtain a dynamic IP address via DHCP. • manual: no address. Static mode is set automatically when an IP address is assigned to the interface.
zone	Zone to which the interface belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]` where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.

Parameter	Description
lldp-profile	Profile to send data using Link Layer Discovery Protocol (LLDP). For more details on configuring profiles, see Configuring LLDP Profiles.
ip-addresses	Assign an IP address to the interface. The IP addresses are specified as **[<ip_address/mask>]** or **[<ip_address/mask> <ip_address/mask>]**. In case of several IP addresses (with space used as the separator), the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
mac	Interface MAC address.
mtu	Specify the MTU size.
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.
rx-ring	Buffer size of the RX ring interface of the adapter type.
tx-ring	Buffer size of the TX ring interface of the adapter type.
dhcp-relay	Settings for the DHCP relay on the interface. You need to specify the following: • **enabled**: enable/disable the relay: ◦ **on** ◦ **off** • **utm-address**: IP address of the UserGate interface on which the relay function is added (possible values: <ip | none>). • **server-address**: addresses of DHCP servers where DHCP requests from clients should be redirected.

To delete an adapter or its parameters, use the following command:

```
Admin@nodename# delete network interface adapter <adapter-name>
```

You can delete the following parameters:

Parameter	Description
ip-addresses	Specified IP address.
dhcp-relay server-address	DHCP server IP address.

To display information about all network adapters, use the following command:

```
Admin@nodename# show network interface adapter
```

To display the adapter information, use the following command:

```
Admin@nodename# show network interface adapter <adapter-name>
```

Configuring a VLAN

VLAN interfaces are configured at the **network interface vlan** level.

To add a new VLAN interface, use the following command:

```
Admin@nodename# create network interface vlan
```

Parameters:

Parameter	Description
enabled	Enable/disable a VLAN interface: • **on** • **off**
description	Interface description.
alias	The interface's alias.
iface-type	Interface type: • **l3**: Layer 3 (you can assign an IP address and use it in firewall, content filtering, and other rules; this is the standard interface operation mode).

Parameter	Description
	• **mirror**: interface works in Mirror mode (it can receive traffic from the network equipment SPAN port to analyze it).
iface-mode	IP address assignment mode: • **dhcp**: obtain a dynamic IP address via DHCP. • manual: no address. Static mode is set automatically when an IP address is assigned to the interface.
tag	VLAN tag. Up to 4094 interfaces can be created.
node-name	Cluster node name where the VLAN is created.
interface	The physical interface on which the VLAN is being created.
zone	Zone to which the interface belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]` where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.

Parameter	Description
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.
ip-addresses	Assign an IP address to the interface. The IP addresses are specified as **[<ip_address/mask>]** or **[<ip_address/mask> <ip_address/mask>]**. In case of several IP addresses (with space used as the separator), the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
mac	Interface MAC address.
mtu	Specify the MTU size.
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.
dhcp-relay	Settings for the DHCP relay on the interface. You need to specify the following: • **enabled**: enable/disable the relay: ◦ **on** ◦ **off** • **utm-address**: IP address of the UserGate interface on which the relay function is added. • **server-address**: addresses of DHCP servers where DHCP requests from clients should be redirected.

To edit an existing VLAN, use the following command:

```
Admin@nodename# set network interface vlan <vlan-name>
```

The parameters available for setting are the same as those for creating a VLAN, except for **tag**, **node-name**, and **interface** (you cannot change these parameter values).

To delete a VLAN interface or its parameters, use the following command:

```
Admin@nodename# delete network interface vlan <vlan-name>
```

You can delete the following parameters:

Parameter	Description
ip-addresses	Specified IP address.
dhcp-relay server-address	DHCP server IP address.

To display information about all VLAN interfaces, use the following command:

```
Admin@nodename# show network interface vlan
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface vlan <vlan-name>
```

Properties of bond interfaces

You configure bond interface properties at the **network interface bond** level.

To create a bond interface, use the following command:

```
Admin@nodename# create network interface bond
```

You need to specify the following parameters:

Parameter	Description
enabled	Enable/disable the interface: • **on** • **off**
interface-name	Enter a number to include in the interface name (for example, if you enter 1 the interface name will be bond1).
description	Interface description.
alias	The interface's alias.
node-name	Cluster node where the bond interface is created.

Parameter	Description
zone	Zone to which the bond belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]` where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.
bonding	Additional bond interface parameters: • **aggr-mode**: bond operation mode. The available options: ◦ **round-robin**: Round robin mode (packets are sent sequentially starting with the first available interface and ending with the last one. This policy is used to provide load balancing and high availability.) ◦ **active-backup**: Active backup mode (only one network interface in the bond will be active. Another slave interface can become active only if the currently active interface fails. With this policy, the MAC address of the bond interface is only visible externally through one network port to avoid problems with the switch. This policy is used to provide high availability).

Parameter	Description
	◦ **xor**: XOR mode (the transmission is allocated among the NICs using the following formula: [(XOR) MOD]. This means that the same NIC sends packets to the same recipients. Optionally, the transmission allocation can also be based on the xmit_hash policy. The XOR policy is used for load balancing and high availability). ◦ **broadcast**: Broadcast mode (broadcasts everything to all network interfaces. This policy is used for high availability). ◦ **802.3ad**: IEEE 802.3ad mode (the default mode supported by most network switches. Creates aggregated groups of NICs with identical speed and duplex settings. When combined like this, all links in the active aggregation participate in transmission as per IEEE 802.3ad. The choice of interface for packet transmission is determined by the policy. By default, the XOR policy is used, with the xmit_hash policy as a possible alternative). ◦ **transmit**: Adaptive transmit load balancing mode (outgoing traffic is distributed depending on the loading of each NIC (determined by the load speed). No additional configuration on the switch is required. The incoming traffic is received by the current network card. If this card fails, another card assumes the MAC address of the failed one). ◦ **load**: Adaptive load balancing mode. Includes the previous policy plus incoming traffic balancing. No additional configuration on the switch is required. The incoming traffic is balanced through ARP negotiation. The driver intercepts ARP responses sent from the local NICs to the outside and overwrites the source MAC address with one of the unique MAC addresses of the NIC in the bond. Thus, different peers use different server MAC addresses. The incoming traffic is balanced sequentially (round-robin) among the interfaces. • **mii-monitoring**: MII monitoring period in milliseconds. Determines how often the link state will be checked for failures. • **down-delay**: delay time (in milliseconds) before an interface is disabled if a connection failure occurs. This option is only valid for MII monitoring (miimon). The parameter value must be a multiple of miimon, • **up-delay**: delay time in milliseconds before deploying the channel if it is detected to be restored. This parameter is only valid with MII monitoring (miimon). The parameter value must be a multiple of miimon,

Parameter	Description
	• **lacp-rate**: interval with which the partner transmits LACPDU packets in 802.3ad mode. Enumerated options: ◦ **slow**: requests that the partner send LACPDU packets every 30 seconds. ◦ **fast**: requests that the partner send LACPDU packets every second. • **failover-mac**: define the assignment type of MAC addresses to bond interfaces in Active backup mode when switching interfaces. Enumerated options: ◦ **disabled**: the same MAC address is set on all interfaces during switching. ◦ **active**: the MAC address on the bond interface will always be identical to that on the currently active slave. The MAC addresses on the backup interfaces are not changed. The MAC address on the bond interface changes during the failover processing. ◦ **follow**: the MAC address on the bond interface will be the same as that on the first slave added to the bond. This MAC is not set on the second and subsequent interfaces while they are in backup mode. That MAC address gets assigned during a failover: when a backup slave interface becomes active, it assumes a new MAC (the one on the bond interface), and the formerly active slave is assigned the MAC that the currently active one used to have. • **xmit-hash**: define a hash policy for sending packets over bond interfaces in XOR or IEEE 802.3ad mode. Enumerated options: ◦ **l2**: use only MAC addresses to generate the hash. With this algorithm, the traffic for a particular network host is always sent over the same interface. This algorithm is compatible with IEEE 802.3ad. ◦ **l2-3**: use both MAC and IP addresses to generate the hash. This algorithm is compatible with IEEE 802.3ad. ◦ l3-4: uses IP addresses and transport layer protocols (TCP or UDP) to generate the hash. This algorithm is not universally compatible with IEEE 802.3ad, as both fragmented and non-fragmented packets can be transmitted within a single TCP or UDP interaction. Fragmented packets lack the source and destination ports. As a result, packets from the same session can reach the recipient in

Parameter	Description
	an order other than the intended one because they are sent via different slaves. • **interface**: interfaces to be bonded.
iface-mode	IP address assignment mode: • **dhcp**: obtain a dynamic IP address via DHCP. • manual: no address. Static mode is set automatically when an IP address is assigned to the interface.
iface-type	The type of interface to be created: • **l3**: a Layer 3 interface • **mirror**: a mirroring interface.
ip-addresses	Assign an IP address to the interface. The IP addresses are specified as **[<ip_address/mask>]** or **[<ip_address/mask> <ip_address/mask>]**. In case of several IP addresses (with space used as the separator), the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
mac	Interface MAC address.
mtu	Specify the MTU size.
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.
dhcp-relay	Settings for the DHCP relay on the interface. You need to specify the following: • **enabled**: enable/disable the relay: ◦ **on** ◦ **off** • **utm-address**: IP address of the UserGate interface on which the relay function is added. • **server-address**: addresses of DHCP servers where DHCP requests from clients should be redirected.

To update an existing bond interface, use the following command:

```
Admin@nodename# set network interface bond <bond-name>
```

The parameters available for setting are the same as those for creating a bond interface, except for **interface-name** and **node-name** (you cannot change the values of these parameters).

To delete a bond interface or its parameters, use the following command:

```
Admin@nodename# delete network interface bond <bond-name>
```

You can delete the following parameters:

Parameter	Description
ip-addresses	Specified IP address.
dhcp-relay server-address	DHCP server IP address.
bonding interface	Bonded interfaces.

To display information about all bond interfaces, use the following command:

```
Admin@nodename# show network interface bond
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface bond <bond-name>
```

Bridge Interface Settings

You configure a bridge at the **network interface bridge** level.

To add a new bridge interface:

```
Admin@nodename# create network interface bridge
```

You need to specify the following parameters:

Parameter	Description
enabled	Enable/disable a bridge: • **on** • **off**
interface-name	Enter a number to include in the interface name (for example, if you enter 1 the interface name will be bridge1).
description	Bridge interface description.
alias	The interface's alias.
node-name	Node name of the cluster where the bridge is created.
zone	Zone to which the bridge belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]` where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.
bridging	

Parameter	Description
	Additional bridge parameters: • **iface-type**: interface mode: ◦ **l2**: Layer 2 (you do not need to assign an IP address or specify routes and gateways for the bridge to work correctly. In this mode, the bridge works at the MAC address level by forwarding packets from one network segment to another. Mail security rules cannot be used in this case; content filtering is available in this mode). ◦ **l3**: Layer 3 (you can assign an IP address and use it in firewall, content filtering, and other rules; this is the standard interface operation mode). • **interface**: interfaces to use to create the bridge. • **stp**: enable/disable STP (Spanning Tree Protocol) for protection against network loops: ◦ **on.** ◦ **off** • **forward-delay**: delay before the bridge switches to the active mode (Forwarding) if the STP is enabled (in seconds). • **max-age**: time after which the STP connection is considered lost (in seconds). • **bypass-pair**: interface pair to use to build the bypass bridge. UserGate HSC support is required.
iface-mode	IP address assignment mode: • **dhcp**: obtain a dynamic IP address via DHCP. • manual: no address. Static mode is set automatically when an IP address is assigned to the interface.
ip-addresses	Assign an IP address to the interface. The IP addresses are specified as **[<ip_address/mask>]** or **[<ip_address/mask> <ip_address/mask>]**. In case of several IP addresses (with space used as the separator), the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
mac	Interface MAC address.
mtu	Specify the MTU size.

Parameter	Description
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.
dhcp-relay	Settings for the DHCP relay on the interface. You need to specify the following: • **enabled**: enable/disable the relay: ◦ **on** ◦ **off** • **utm-address**: IP address of the UserGate interface on which the relay function is added. • **server-address**: addresses of DHCP servers where DHCP requests from clients should be redirected.

To update an existing bridge interface, use the following command:

```
Admin@nodename# set network interface bridge <bridge-name>
```

The parameters available for setting are the same as those for creating a bridge, except for **interface-name** and **node-name** (you cannot change the values of these parameters).

To delete a bridge interface or its parameters, use the following command:

```
Admin@nodename# delete network interface bridge <bridge-name>
```

You can delete the following parameters:

Parameter	Description
ip-addresses	Specified IP address.
dhcp-relay server-address	DHCP server IP address.

To display information about all bridge interfaces, use the following command:

```
Admin@nodename# show network interface bridge
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface bridge <bridge-name>
```

PPPoE configuration

PPPoE is configured at the **network interface PPPoE** level.

To create a PPPoE interface, use the following command:

```
Admin@nodename# create network interface pppoe
```

Parameters:

Parameter	Description
enabled	Enable/disable a PPPoE interface: • **on** • **off**
interface-name	Enter a number to include in the interface name (for example, if you enter 1 the interface name will be ppp1).
description	PPPoE interface description.
alias	The interface's alias.
node-name	Cluster node name where the interface is created.
zone	Zone to which the interface belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]`

Parameter	Description
	where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.
config	Additional PPPoE interface parameters: • **interface**: interface where the PPPoE interface is created. • **login**: login name for PPPoE connection. • **password**: password for PPPoE connection. • **persist-connection**: automatic reconnection in case of connection failure: ◦ **on** ◦ **off** • **auth-type**: authorization type: ◦ **CHAP**. ◦ **PAP**. • **holdoff**: time period (in seconds) to restart the connection after it was broken. • **default-route**: use the PPPoE interface as the default route: ◦ **on** ◦ **off** • **echo-interval**l: interval to check the connection. • **echo-failure**: number of LCP echo failures after which UserGate assumes there is no connection and drops it. • **providers-dns**: use DNS servers provided by the ISP: ◦ **on** ◦ **off** • **connection-attempts**: number of unsuccessful connection attempts, after which auto-connection attempts will stop. • **service-name**: specify the service name if provided by the ISP.

Parameter	Description
mtu	Specify the MTU size. Set by default to a value of 1492 bytes compatible with the standard Ethernet frame size.
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.

To update an existing PPPoE interface, use the following command:

```
Admin@nodename# set network interface pppoe <pppoe-name>
```

The parameters available for setting are the same as those for creating an interface, except for **interface-name** (you cannot change this parameter's value).

To delete a PPPoE interface, use the following command:

```
Admin@nodename# delete network interface pppoe <pppoe-name>
```

To display information about all PPPoE interfaces, use the following command:

```
Admin@nodename# show network interface pppoe
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface pppoe <pppoe-name>
```

Configuring a VPN device

You configure VPN devices at the **network interface vpn** level.

To create a VPN device, use the following command:

```
Admin@nodename# create network interface vpn
```

Parameters:

Parameter	Description
enabled	Enable/disable a VPN interface: • **on** • **off**
interface-name	Enter a number to include in the interface name (for example, if you enter 1 the interface name will be tunnel1).
description	VPN interface description.
alias	The interface's alias.
zone	Zone to which the interface belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]` where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.
iface-mode	IP address assignment mode: • **dhcp**: obtain a dynamic IP address via DHCP. • manual: no address.

Parameter	Description
	If the interface is to be used for receiving VPN connections (Site-2-Site VPN or Remote access VPN), a static IP address must be used. Static mode is set automatically when an IP address is assigned to the interface. To use an interface as a client, select the dynamic mode.
ip-addresses	Assign an IP address to the interface. The IP addresses are specified as **[<ip_address/mask>]** or **[<ip_address/mask> <ip_address/mask>]**. In case of several IP addresses (with space used as the separator), the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
mtu	Specify the MTU size for the selected interface.
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.

To update an existing VPN interface, use the following command:

```
Admin@nodename# set network interface vpn <vpn-name>
```

The parameters available for setting are the same as those for creating an interface, except for **interface-name** (you cannot change this parameter's value).

To delete a VPN interface or its parameters, use the following command:

```
Admin@nodename# delete network interface vpn <vpn-name>
```

You can delete the following parameters: **ip-addresses**.

To display information about all VPN interfaces, use the following command:

```
Admin@nodename# show network interface vpn
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface vpn <vpn-name>
```

Configuring tunnels

You create and configure tunnels at the **network interface tunnel** level.

To create a tunnel, use the following command:

```
Admin@nodename# create network interface tunnel
```

Parameters:

Parameter	Description
enabled	Enable/disable the tunnel: • **on** • **off**
interface-number	Enter a number to include in the tunnel name (for example, if you enter 1 the interface name will be gre1).
description	Tunnel description.
alias	The interface's alias.
node-name	Cluster node where the tunnel is created.
zone	Zone to which the interface belongs.
link-info	Settings for network interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]` where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and

Parameter	Description
	value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
mtu	The MTU size for the selected interface.
mss	MSS size (available starting from software version 7.3.x). Correct values: • 0; • from 4 to the entered MTU value minus 40 (for VXLAN); • from 4 to the entered MTU value minus 60 (for IPIP); • From 4 to the specified MTU value minus 64 (for GRE).
ip-addresses	The IP address assigned to the tunnel interface. The IP addresses are specified as **[<ip_address/mask>]** or **[<ip_address/mask> <ip_address/mask>]**. In case of several IP addresses (with space used as the separator), the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
local-ip	The local address of the Point-to-Point interface.
remote-ip	The remote address of the Point-to-Point interface.
mode	The tunnel operation mode: • **gre**: GRE (a network packet tunneling protocol developed by Cisco Systems. Its main purpose is to encapsulate network layer packets into IP packets. The IP protocol number is 47. • **ipip**: IPIP (an IP tunneling protocol that encapsulates one IP packet in another IP packet. Encapsulating one IP packet in another IP packet adds an external header with the Source IP which is the entry point into the tunnel and the Destination IP which is the exit point from the tunnel). • **vxlan**: VXLAN (tunneling protocol from Layer 2 Ethernet frames to UDP packets, port 4789).
vxlan-id	The VXLAN ID. Relevant only for a VXLAN tunnel.

To edit an existing tunnel parameters, use the following command:

```
Admin@nodename# set network interface tunnel <tunnel-name>
```

The parameters available for setting are the same as those for creating an interface, except for **interface-number** and **node-name** (you cannot change these parameter values).

To delete a tunnel interface or its parameters, use the following command:

```
Admin@nodename# delete network interface tunnel <tunnel-name>
```

You can delete the following parameters: **ip-addresses**.

To display information about all tunnels, use the following command:

```
Admin@nodename# show network interface tunnel
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface tunnel <tunnel-name>
```

Properties of loopback interfaces

You create and configure a loopback interface at the **network interface loopback** level.

To create an interface, use the following command:

```
Admin@nodename# create network interface loopback
```

Parameters:

Parameter	Description
enabled	Enable/disable the interface: • **on** • **off**
interface-name	The interface name.
description	Network interface description.
alias	The interface's alias.
ip-addresses	Assign an IP address to the interface. The IP addresses are specified as **[<ip_address/mask>]**, the subnet mask is entered in the decimal format. **Important!** Make sure to separate the square brackets with spaces on both sides.
iface-mode	IP address assignment mode: • **dhcp**: obtain a dynamic IP address via DHCP. • manual: no address. Static mode is set automatically when an IP address is assigned to the interface.
lldp-profile	Profile to send data using Link Layer Discovery Protocol (LLDP). For more details on configuring profiles, see Configuring LLDP Profiles.
zone	Zone to which the interface belongs.
link-info	Settings for interface parameters: • **bc_forwarding**: control forwarding the directed broadcast packets arriving at the specified interface. • **proxy_arp**, **proxy_arp_vlan**: Proxy ARP mechanism. With **proxy_arp**, UserGate will respond to ARP requests for addresses outside the interface's network; with **proxy_arp_vlan**, UserGate will respond to ARP requests for addresses that belong to the interface's network. To specify them, use the following format: `Admin@nodename# create network interface <iface-type> … link-info [ key/value ]`

Parameter	Description
	where key is the parameter name. which can include lowercase Latin letters (a-z) and underscore (_), and value is the parameter value. Parameter values can only be integers. For example, use proxy_arp/1 to enable the Proxy ARP mechanism and proxy_arp/0 to disable it. The link-info field is displayed only when adding parameters. **Important!** You cannot delete the specified parameters.
netflow-profile	The Netflow profile to send statistical data to the Netflow collector. For more details on Netflow profile settings, see Configuring Netflow Profiles.
node-name	Cluster node where the interface is created.
mac	Interface MAC address.
mtu	Specify the MTU size.
mss	Specifying the MSS size: 0, or from 4 to the specified MTU value minus 40.
dhcp-relay	Settings for the DHCP relay on the interface. You need to specify the following: • **enabled**: enable/disable the relay: ◦ **on** ◦ **off** • **utm-address**: IP address of the UserGate interface on which the relay function is added (possible values: <ip | none>). • **server-address**: addresses of DHCP servers where DHCP requests from clients should be redirected.

To edit an existing interface, use the following command:

```
Admin@nodename# set network interface loopback <interface-name>
```

The parameters available for setting are the same as those for creating a loopback interface, except for **node-name** and **interface** (you cannot change these parameter values).

To delete a loopback interface or its parameters, use the following command:

```
Admin@nodename# delete network interface loopback <interface-name>
```

You can delete the following parameters:

Parameter	Description
ip-addresses	Specified IP address.
dhcp-relay	DHCP server IP address.

To display information about all loopback interfaces, use the following command:

```
Admin@nodename# show network interface loopback
```

To display information about a single interface, use the following command:

```
Admin@nodename# show network interface loopback <interface-name>
```

Gateways

This section is located at the **network gateway** level.

To add a new gateway, use the following command:

```
Admin@nodename# create network gateway
```

Available parameters:

Parameter	Description
enabled	Enable/disable the gateway: • **on** • **off**
name	Gateway name.
description	Gateway description.

Parameter	Description
interface	Interface used to access the Internet: • **Select a specific port (port0, port1, port2, etc.);** • **auto: after selecting this option, the port will be detected automatically.**
virtual-router	Select a virtual router for which the gate is configured.
ip	Gateway IP address.
node-name	Select the cluster node for which the gateway is configured.
weight	Gateway weight (the greater the weight, the greater the share of traffic goes through the gateway).
balancing	Balancing mode: all traffic to the Internet will be distributed between the gateways according to their weights: • **on** • **off**
default	Use this gateway as the default gateway: • **on** • **off**

To update gateway parameters, use the following command:

```
Admin@nodename# set network gateway <gateway-name>
```

You can use the same set of parameters as when creating a gateway.

To delete a gateway, use the following command:

```
Admin@nodename# delete network gateway <gateway-name>
```

To display information about all gateways, use the following command:

```
Admin@nodename# show network gateway
```

To display information about a single gateway, use the following command:

```
Admin@nodename# show network gateway <gateway-name>
```

DHCP

This section is located at the **network dhcp** level.

To create a DHCP subnet, use the following command:

```
Admin@nodename# create network dhcp
```

Parameters:

Parameter	Description
enabled	Enable/disable the use of this IP address range: • **on** • **off**
name	Subnet name.
description	Subnet description.
interface	Interface of the server which will assign IP addresses from the range being created.
ip-range	The IP address range assigned to DHCP clients. Format: <IP_start-IP_end>.
mask	The subnet mask assigned to DHCP clients.
expiration-time	The duration in seconds for which IP addresses are assigned.
domain	The domain name assigned to DHCP clients.
gateway	The gateway IP address assigned to DHCP clients.
dns-servers	The DNS server IP addresses assigned to DHCP clients.

Parameter	Description
reserved-hosts	The MAC addresses and the associated IP addresses: • **mac**: MAC address. • **ip**: IP address associated with the MAC address. • **hostname**: name of the host.
ignored-mac	List of MAC addresses ignored by the DHCP server.
pxe-boot-ip	PXE boot server IP.
pxe-boot-filename	PXE boot filename.
options	Option number and value: • **code**: DHCP option number. • **values**: option value.

To update an existing DHCP subnet, use the following command:

```
Admin@nodename# set network dhcp <dhcp-name>
```

The parameters available for settings are the same as those used when creating a subnet.

To delete a subnet, use the following command:

```
Admin@nodename# delete network dhcp <dhcp-name>
```

You can also delete individual DHCP subnet parameters:

- **dns-servers**.
- **ignored-mac**.
- **reserved-hosts** (specify all three values: **mac**, **ip**, and **hostname**)
- **options** (specify both values: **code** and **values**).

To display information about all subnets created, use the following command:

```
Admin@nodename# show network dhcp
```

To display information about a specific DHCP subnet, use the following command:

```
Admin@nodename# show network dhcp <dhcp-name>
```

DNS Configuration

This section is located at the **network dns** level.

Settings for System DNS servers

You configure system DNS servers at the **network dns system-dns-servers** level.

To add new DNS servers or update the list of existing ones, use the following commands:

```
Admin@nodename# set network dns system-dns-servers ip [ <ip> <ip> ... ]
```

To delete the entire list of DNS server addresses, use the following command:

```
 Admin@nodename# delete network dns system-dns-servers
```

To delete individual servers, use the following command:

```
 Admin@nodename# delete network dns system-dns-servers ip [ <ip>
<ip> ... ]
```

To display the list of system DNS servers, use the following command:

```
 Admin@nodename# show network dns system-dns-servers
```

DNS proxy settings

You configure DNS proxies at the **network dns proxy-settings** level.

To edit DNS proxy settings, use the following command:

```
Admin@nodename# set network dns proxy-settings
```

Add the parameters you want to change:

Parameter	Description
filtering	DNS request filtering: • **on** • **off**
caching	Cache DNS responses: • **on** • **off**
limit	Limit the number of DNS queries per second for each user (default value: 100).
max-ttl	Maximum possible time-to-live for DNS records.
recursive	Perform recursive DNS queries: • **on** • **off**
dns-timeout	Time to the next attempt to query a DNS server (in milliseconds).
a-aaaa-unknown	Respond only to requests for A and AAAA records from unknown users. This effectively blocks attempts to establish a VPN over the DNS protocol: • **on** • **off**
retries	Number of attempts to send a DNS request.
factory-defaults	

Parameter	Description
	Reset the values of the selected parameter (parameters shown in this table) or all parameters (**all**) to factory defaults.

Example command to edit DNS-proxy parameters:

```
Admin@nodename# set network dns proxy-settings limit 10 dns-timeout 10
```

To display DNS proxy settings, use the following command:

```
Admin@nodename# show network dns proxy-settings
```

Configuring DNS rules

> **Please note!**
> **DCFW cannot be used as a DNS-server. It is supposed to work in the role of DNS proxy only. Specifying the DCFW address in the DNS rules may cause a DNS loop formation followed by all the ensuing consequences.**

DNS rules are configured at the **network dns rules** level using the UPL syntax. For more details on the command structure, see UserGate Policy Language.

DNS rule parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable the rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	The name of the rule. Example: **name("DNS rule example")**.
desc	DNS proxy rule description. Example: **desc("DNS rule example set via CLI")**.

Parameter	Description
url.domain	List of domains to which you want to redirect. You can use an asterisk (*) to specify a domain template. To specify a list of domains: **url.domain = "*.example.com"**.
dns_server	List of DNS server IP addresses to which requests for the specified domains should be forwarded. To specify a server: **dns_server(1.2.3.4)**.

Example command to create a DNS rule using UPL:

```
Admin@nodename# create network dns rules 1 upl-rule OK \
...url.domain = "*.example.com" \
...dns_server(1.2.3.4) \
...name("DNS rule example") \
...desc("DNS rule example description over CLI") \
...enabled(true) \
...
Admin@nodename#
Admin@nodename# show network dns rules

% ---------------- 1 -----------------
OK \
    url.domain = "*.example.com" \
    dns_server(1.2.3.4) \
    desc("DNS rule example description over CLI") \
    enabled(true) \
    id("0f83e1bb-0aa5-4f42-8eeb-9c4ffa30c04a") \
    name("DNS rule example")
```

Configuring DNS proxy static records

This section is located at the **network dns static-records** level.

To add a static DNS record, use the following command:

```
Admin@nodename# create network dns static-records
```

Specify the parameters:

Parameter	Description
enabled	Enable/disable static record usage: • **on** • **off**
name	Record name.
description	DNS record description.
domain	Static record FQDN (Fully Qualified Name), e.g. www.example.com.
dns-a-records	List of IP addresses the UserGate server will return when this FQDN is queried.

Command

```
Admin@nodename# show network dns static-records
```

displays information about all existing static DNS records. To display information about a specific record, use the following command:

```
Admin@nodename# show network dns static-records <static-record-name>
```

Example of creating a static DNS record:

```
Admin@nodename# create network dns static-records name "Test DNS static
record" description "Test DNS static record description" enabled on
domain example.com dns-a-records [ 10.10.0.100 ]
Admin@nodename#
Admin@nodename# show network dns static-records

Test DNS static record
    name              : Test DNS static record
    description       : Test DNS static record description
    domain            : example.com
```

```
    dns-a-records    : 10.10.0.100
    enabled          : on
```

To edit information about static DNS records:

```
Admin@nodename# set network dns static-records <static-record-name>
```

The set of parameters available to change is the same as those for the **create** command.

An example of editing a previously created static DNS record:

```
Admin@nodename# set network dns static-records "Test DNS static record"
dns-a-records [ 10.10.0.101 ]
Admin@nodename# show network dns static-records "Test DNS static
record"

name             : Test DNS static record
description      : Test DNS static record description
domain           : example.com
dns-a-records    : 10.10.0.100; 10.10.0.101
enabled          : on
```

To delete a static record, use the following command:

```
 Admin@nodename# delete network dns static-records <static-record-name>
```

You can also delete only the **dns-a-records** parameter values from the static record.

An example of deleting the value of the **dns-a-records** parameter in a previously created record and deleting the entire static DNS record.

```
Admin@nodename# delete network dns static-records "Test DNS static
record" dns-a-records [ 10.10.0.101 ]
Admin@nodename# show network dns static-records "Test DNS static
record"

name             : Test DNS static record
```

```
description      : Test DNS static record description
domain           : example.com
dns-a-records    : 10.10.0.100
enabled          : on

Admin@nodename# delete network dns static-records "Test DNS static
record"
Admin@nodename# show network dns static-records


Admin@nodename#
```

Configuring Virtual Routers

This section describes how to configure static routes, OSPF, BGP, and RIP dynamic routing protocols, and multicast routing using the CLI (the configuration is discussed in the respective sections). These settings are applied at the **network virtual-router** level.

Commands used to configure general settings of virtual routers are listed below.

To add a new virtual router, use the following command:

```
Admin@nodename# create network virtual-router <parameters>
```

Specify the parameters:

Parameter	Description
name	Virtual router unique name.
description	Virtual router description.
node-name	Select a UserGate node where the virtual router will be created (if a cluster exists).
interfaces	Interfaces to use on this virtual router. You cannot add interfaces already added to other virtual routers. An interface can belong to only one virtual router. All types of interfaces, including physical, virtual (VLAN), bond, VPN, and others can be added to a virtual router.

To display information about a virtual router, use the following command:

```
Admin@nodename# show network virtual-router <virtual-router-name>
```

Example of creating a virtual router:

```
Admin@nodename# create network virtual-router name test_router
description "Test virtual router" interfaces [ port2 ]
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2
...
```

To edit virtual router parameters, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name>
```

The parameters available to update are the same as those for the **create** command, except for:

- **name**.
- **node-name**.

Example of editing virtual router parameters:

```
Admin@nodename# set network virtual-router test_router interfaces
[ port3 ]
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
...
```

To delete a virtual router, use the following command:

```
Admin@nodename# delete network virtual-router <virtual-router-name>
```

Configuring static routes

To add a new static route, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name> routes
new
```

Specify the parameters:

Parameter	Description
enabled	Enable/disable usage of a static route: • **on** • **off**
name	Route name.
description	Route description.
type	Route type: • **unicast**: the standard route type. Forwards the traffic destined for the specified address via the specified gateway. • **unreachable**: drops the traffic. and sends the "Host unreachable" (type 3 code 1) ICMP message to the source. • **prohibit**: drops the traffic. and sends the "Host unreachable" (type 3 code 13) ICMP message to the source. • **blackhole**: drops the traffic without informing the source that the data did not reach the recipient.
destination-ip	IP address of the destination subnet, format: <ip/mask>.
gateway	IP address of the gateway through which the specified subnet will be reachable. The IP address must be reachable from the UserGate server.

Parameter	Description
interface	Interface through which the route is added.
metric	Route metric. If there is more than one route to this network: the lower the metric, the higher the priority of the route.

Example of adding a static route:

```
Admin@nodename# set network virtual-router test_router routes new name
"Test static route" description "Test static route description"
destination-ip 192.168.200.0/24 gateway 192.168.100.100 interface port3
type unicast metric 1 enabled on
Admin@nodename#
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
routes              :
    Test static route
        name              : Test static route
        enabled           : on
        description       : Test static route description
        destination-ip    : 192.168.200.0/24
        gateway           : 192.168.100.100
        interface         : port3
        metric            : 1
...
```

To change the parameters of an existing static route, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name> routes
<static-route-name>
```

The parameters available to change are listed in the table above.

Example of editing a static route:

```
Admin@nodename# set network virtual-router test_router routes "Test
static route" metric 10
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
routes              :
    Test static route
        name              : Test static route
        enabled           : on
        description       : Test static route description
        destination-ip    : 192.168.200.0/24
        gateway           : 192.168.100.100
        interface         : port3
        metric            : 10
...
```

To delete a static route, use the following command:

```
Admin@nodename# delete network virtual-router <virtual-router-name>
routes <static-route-name>
```

Example of deleting a static route:

```
Admin@nodename# delete network virtual-router test_router routes "Test
static route"
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
routes              : []
...
```

To display static routes, use the following command:

```
 Admin@nodename# show network virtual-router <virtual-router-name>
routes
```

OSPF Configuration

To configure OSPF using CLI, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name> ospf
```

Provide the following OSPF router parameters:

Parameter	Description
enabled	Enable/disable an OSPF router: • **on** • **off**
router-id	Router IP address. Must be unique and specified in IPv4 format (for convenience, it can match one of the IP addresses assigned to the UserGate network interfaces that belong to this virtual router). If the OSPF is disabled (**enabled off**), the **router-id** value can be deleted (**none**).
metric	Redistributed route metric.
default-originate	Notify other routers that this router has a default route configured: • **on** • **off**
interfaces	Select one of the existing interfaces on which OSPF will run. Only the interfaces belonging to this virtual router are available for selection. To add an interface or change parameters for an existing interface, use the following commands:

Parameter	Description
	`` Admin@nodename# set network virtual-router <virtual-router-name> ospf interfaces new Admin@nodename# set network virtual-router <virtual-router-name> ospf interfaces <interface-name> ``` Next, specify the following parameters: • **enabled <on | off>**: enable/disable the interface. • **interface**: name of the interface in this virtual router. • **description**: interface description. • **bfd**: Add a bfd profile (Bidirectional Forwarding Detection). Bfd profiles are created in the element library, read more in the Configuring Libraries section. • **cost**: interface link cost. This value is reported in the LSA (link-state advertisement) to the neighboring routers which use it to compute the shortest path. Default value: 1. • **priority**: an integer from 0 to 255. The higher the value, the higher the probability that this router will become the network's designated router for sending out LSAs. A value of 0 excludes the router from being designated. Default value: 1. • **network-type**: select a network type to optimize the adjacency establishment process. Available values: ◦ **none**: not specified ◦ **bc**: broadcast ◦ **ptm**: point to multipoint ◦ **ptp**: point to point • **passive-mode <on | off>**: enable/disable the passive operating mode of the interface, in which routing protocol update packets are prohibited from being sent through the interface. • **hello-interval**: time between sending hello packets (in seconds). This should be the same for all routers in an autonomous system. The default value is 10 seconds. • **dead-interval**: time after which the router is considered offline (in seconds). The time is counted from the moment of receiving the last hello packet from the neighboring router. The default value is 40 seconds. • **retransmit-interval**: time before the LSA packet is retransmitted (in seconds). The default value is 5 seconds.

Parameter	Description
	• **transmit-delay**: approximate time required to deliver link state updates to neighbor routers (in seconds). The default value is 1 second. • **authentication**: authentication type. Available values: ◦ **enabled <on | off>**: enable / disable mandatory authentication for each OSPF message received by the router. Authentication is normally used to prevent the injection of a fake route from illegitimate routers. ◦ **auth-type**: select the authentication type as **plain** (transmit the key as plain text to authenticate routers) or **digest** (use MD5 hash of the key to authenticate OSPF packets). ◦ **md5**: the key ID. ◦ **key**: the key. A key can only contain Latin letters, numbers, and the underscore. Maximum length: 16 characters.
areas	Configuring the OSPF area. To add a new area or change parameters for an existing one, use the following commands: `Admin@nodename# set network virtual-router <virtual-router-name> ospf areas new` `Admin@nodename# set network virtual-router <virtual-router-name> ospf areas <area-name>` Next, specify the following parameters: • **enabled <on | off>**: enable/disable the area. • **name**: area name. • **description**: area description. • **cost**: cost of the LSAs announced in the stub area. • **area-id**: zone ID (area ID). The ID can be specified in decimal format or IP address record format. The area ID must match to establish an OSPF adjacency. • **auth-type**: authentication type. Available values: ◦ **none**: do not require OSPF packet authentication. ◦ **plain**: transmit the key as plain text to authenticate OSPF packets. The key specified in the interface settings is used.

Parameter	Description
	◦ **digest**: use MD5 hash of the key to authenticate OSPF packets. The key specified in the interface settings is used. The interface-level authentication takes precedence over zone-level authentication. • **area-type**: OSPF area type. Available types: ◦ **normal**: normal zone, created by default. This zone receives link updates, summary routes, and external routes. ◦ **nssa**: a Not-So-Stubby Area defines an additional LSA type, which is LSA type 7. A boundary router (ASBR) can be located in the NSSA zone. ◦ **stub**: a stub area. Does not receive information on routes external to the autonomous system but receives routes from other areas. If routers from a stub area need to send information outside of the autonomous system, they use the default route. An ASBR cannot reside in a stub area. • **no-summary**: allow/deny summarized routes to be injected into stub zone area types: ◦ **on** ◦ **off** • **interfaces**: select the OSPF interfaces on which this area will be available. • **virtual-links**: this is a special type of connection that makes it possible, for example, to interconnect a partitioned area or connect an area to the backbone area via another area. It is configured between two ABRs. Routers can transmit OSPF packets encapsulated in IP packets over such links. This mechanism is used as a temporary solution or as a backup in case the primary connections fail. You can specify the IDs of the routers available via this zone.
redistribute	OSPF route redistribution: • **connected**: redistribute routes to the networks directly connected to UserGate • **kernel**: redistribute routes added by the administrator.

To display a OSPF configuration of a virtual router, use the following command:

```
Admin@nodename# show network virtual-router <virtual-router-name> ospf
```

Examples of OSPF configuring in a virtual router:

```
Admin@nodename# set network virtual-router test_router  ospf router-id
192.168.100.3 areas new area-id 1 area-type normal name "New OSPF area"
enabled on interfaces [ ]
...
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
routes              : []
ospf                :
    router-id            : 192.168.100.3
    enabled              : off
    default-originate    : off
    metric               : None
    areas                :
        New OSPF area
            name             : New OSPF area
            enabled          : on
            cost             : 1
            area-id          : 1
            area-type        : normal
            no-summary       : off

    interfaces           : []
...
```

To delete OSPF settings, use the following command:

```
 Admin@nodename# delete network virtual-router <virtual-router-name>
ospf <parameter>
```

You can delete the following parameters:

- **interface**
- **area**

Configuring BGP

To configure BGP (Border Gateway Protocol) dynamic routing protocol on a virtual router, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name> bgp
```

Specify the parameters:

Parameter	Description
enabled	Enable/disable an OSPF router: • **on** • **off**
router-id	Router IP address. Must match one of the IP addresses assigned to the UserGate network interfaces that belong to this virtual router. If the BGP is disabled (**enabled off**), the **router-id** value can be deleted (**none**).
asn	An autonomous system is a system of IP networks and routers managed by one or more operators that have a single routing policy. The autonomous system number identifies the router as belonging to that system.
multiple-path	Enable/disable traffic balancing to routes with the same cost: • **on** • **off**
redistribute	BGP route redistribution: • **connected**: redistribute routes to the networks directly connected to UserGate • **kernel**: redistribute routes added by the administrator. • **ospf**: redistribute routes received via the OSPF protocol.

Parameter	Description
networks	A list of networks that belong to this autonomous system. Format: <ip/mask>.
routemaps	Routemaps are used to manage routing tables and specify the match conditions under which routes are passed between domains. To create a routemap or change parameters for an existing routemap, use the following commands: `Admin@nodename# set network virtual-router <virtual-router-name> bgp routemaps new` `Admin@nodename# set network virtual-router <virtual-router-name> bgp routemaps <routemap-name>` Routemap parameters: • **name**: routemap name. • **description**: routemap description. • **action**: action: ◦ **allow**: allow data that matches the routemap conditions to pass through ◦ **block**: deny data that matches the routemap conditions to pass through. • **match-by**: match condition to apply a routemap. Match by: ◦ **ip**: IP address. ◦ **aspath**: AS path. ◦ **community**: Community. • **next-hop**: set next hop value for filtered routes to the specified IP address. • **weight**: set the weight for filtered routes to the specified value. • **metric**: set the metric for filtered routes to the specified value. • **preference**: set the preference for filtered routes to the specified value. • **as-prepend**: set the AS-prepend value, which is a list of autonomous systems being added for this route. • **community**: set the BGP community value for filtered routes.

Parameter	Description
	• **append-community**: append community. • **ip-match**: add all required IP addresses when selecting IP address matching. • **as-path-match**: add all required autonomous network numbers when selecting AS path matching. POSIX 1003.2 regular expressions are allowed, supplemented by the underscore (_) character that is interpreted as: ◦ A space ◦ A comma ◦ Start of line ◦ End of line ◦ AS set delimiter { and } ◦ AS confederation delimiter (and). ◦ **community-match**: add the strings of all desired BGP communities when matching by Community is selected.
filters	Filters allow you to filter routes when redistributing. To create a filter or change parameters for an existing one, use the following commands: `Admin@nodename# set network virtual-router <virtual-router-name> bgp filters new` `Admin@nodename# set network virtual-router <virtual-router-name> bgp filters <filter-name>` Parameters: • **name**: the filter name. • **description**: the filter description. • **action**: action: ◦ **allow**: allow data that matches the routemap conditions to pass through ◦ **block**: deny data that matches the routemap conditions to pass through. • **filter-by**: conditions on application of the filter. The following actions are available: ◦ **ip**: filter by the IP address. ◦ **aspath**: filter by the AS path.

Parameter	Description
	• **ip-filter**: add all desired IP addresses when IP address filtering is selected. The addresses can be specified in the following formats: ◦ 10.0.0.0/8 for the 10.0.0.0/8 subnet only ◦ 10.0.0.0/8:11 for routes where the first octet is 10 and the prefix is from 8 to 11 ◦ 10.0.0.0/8:11:13 for routes where the first octet is 10 and the prefix is from 11 to 13. • **as-path-filter**: add all required autonomous network numbers when selecting filtering by AS path.
neighbors	BGP neighbors. To add new neighbors or change data for existing ones, use the following commands: `Admin@nodename# set network virtual-router <virtual-router-name> bgp neighbors new` `Admin@nodename# set network virtual-router <virtual-router-name> bgp neighbors <host-ip>` Parameters: • **enabled**: enable/disable use of the neighbor: ◦ **on** ◦ **off** • **description**: BGP neighbor description. • **host**: neighbor IP address. • **remote-asn**: neighbor's autonomous system number. • **weight**: weight of routes received from this neighbor. • **ttl**: maximum allowed hop number to this neighbor. • **allowas-in**: allows receiving and processing routes even if the router detects its own autonomous system number on the AS Path in the aggregation route. ◦ **on** ◦ **off** • **allowas-in-number**: how many times the autonomous BGP neighbor's system number can be included in the AS Path. Available values: from 0 to 10 (0 is the origin). • **bfd**: Add a bfd profile (Bidirectional Forwarding Detection). Bfd profiles are created in the element library, read more in the Configuring Libraries section.

Parameter	Description
	• **next-hop-self**: if the neighbor is a BGP, replace the next-hop-self value with its own IP address: ◦ **on** ◦ **off** • **ebgp-multihop**: the connection to this BGP neighbor is not direct (more than one hop): ◦ **on** ◦ **off** • **route-reflector-client**: determine if a BGP neighbor is a Route reflector client: ◦ **on** ◦ **off** • **soft-reconfiguration**: use soft reconfiguration (without disconnecting) to update the configuration: ◦ **on** ◦ **off** • **default-originate**: announce the default route to a neighbor: ◦ **on** ◦ **off** • **send-community**: redirect the community to BGP neighbors. ◦ **on** ◦ **off** • **enable-auth**: enable/disable authentication for the neighbor. ◦ **on** ◦ **off** • **password**: the neighbor authentication password. • **filter-in**: restrict routing information received from neighbors. • **filter-out**: restrict routing information announced to neighbors. • **routemap-in**: restrict routing information that BGP receives from neighbors. • **routemap-out**: restrict routing information that BGP sends to neighbors.

To display BGP configuration in a virtual router, use the following command:

```
Admin@nodename# show network virtual-router <virtual-router-name> bgp
```

Example command to configure BGP in a virtual router:

```
Admin@nodename# set network virtual-router test_router bgp router-id
192.168.95.224 asn 1 networks [ 192.168.100.0/24 ] redistribute
[ connected kernel ]
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
...
bgp                 :
    enabled          : off
    asn              : 1
    router-id        : 192.168.95.224
    redistribute     : connected; kernel
    multiple-path    : off
    networks         : 192.168.100.0/24
    routemaps        : []
    neighbors        : []
    filters          : []
...
```

To delete BGP router parameters, use the following command:

```
 Admin@nodename# delete network virtual-router <virtual-router-name>
bgp <parameter>
```

You can delete the following parameters:

- Addresses of networks that belong to this autonomous system: **networks**.
- Conditions on application of routemap: **routemaps <routemap-name> ip-match | community-match | as-path-match**.
- Condition on application of filters: **filters <filter-name> ip-filter | as-path-filter**.

- BGP neighbors and routemap filters: **neighbors <host-ip> filter-in | filter-out | routemap-in | routemap-out**.
- BGP route redistribution options: **redistribute [connected | kernel]**.

RIP Configuration

To configure RIP (Routing Information Protocol) on a virtual router, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name> rip
```

Specify the parameters:

Parameter	Description
enabled	Enable/disable an RIP router: • **on** • **off**
version	RIP protocol version: • **1** • **2** Usually, the 2nd version of the protocol is used.
metric	RIP metric. Default value: 1; max value: 15. A value of 16 is considered infinite.
distance	The cost of routes received using the RIP protocol. Default value for RIP protocol: 120. This is used for route selection when routes can be received using multiple methods (OSPF, BGP, static).
originate	Sends itself as the router by default.
networks-cidr	Specify the network as a CIDR. Format: <ip/mask>.
networks-interface	Specify the network interface from which to send route information updates. Provide interfaces that belong to the virtual router.
redistribute	

Parameter	Description
	Route redistribution: • **connected**: redistribute routes to other RIP routers to the networks directly connected to UserGate: ◦ <metric>: metric value; available values: from 0 to 16 ◦ **off** • **static**: redistribute static routes to other static router. ◦ <metric>: metric value; available values: from 0 to 16 ◦ **off** • **kernel**: redistribute administrator added routes to other RIP routers: ◦ <metric>: metric value; available values: from 0 to 16 ◦ **off** • **ospf**: redistribute routes received via OSPF to other RIP routers: ◦ <metric>: metric value; available values: from 0 to 16 ◦ **off** • **bgp**: redistribute routes received via BGP to other RIP routers: ◦ <metric>: metric value; available values: from 0 to 16 ◦ **off**
interfaces	Configure interfaces where the RIP protocol is supported. The interfaces should be added to the virtual router. To add new interfaces or change data for existing ones, use the following commands: `Admin@UGOS# set network virtual-router <virtual-router-name> rip interfaces new` `Admin@UGOS# set network virtual-router <virtual-router-name> rip interfaces <interface-name>` Parameters: • **interface**: select the interface. • **send-version**: the RIP protocol version that the router will send. Available values: ◦ **0** ◦ **1**

Parameter	Description
	◦ **2** ◦ **3** • **receive-version**: the RIP protocol version that the router will receive. Available values: ◦ **0** ◦ **1** ◦ **2** ◦ **3** • **password**: the authorization string that will be sent and received in RIP packets. All routers participating in RIP information exchange must have an identical password. • **split-horizone**: a routing loop avoidance method where the router does not redistribute network information through the interface on which the update arrived. ◦ **on** ◦ **off** • **poisoned-reverse**: a routing loop avoidance method where the router sets the route cost to 16 and sends it to the neighbor from which it was received. ◦ **on** ◦ **off** • **passive-mode**: an interface mode in which the interface receives RIP updates but does not send them. ◦ **on** ◦ **off**

To display RIP configuration in a virtual router, use the following command:

```
Admin@nodename# show network virtual-router <virtual-router-name> rip
```

Example command to configure RIP in a virtual router:

```
Admin@nodename# set network virtual-router test_router rip version 2
originate on
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
```

```
node-name           : node_1
interfaces          : port2; port3
...
rip                 :
    enabled               : off
    distance              : 120
    metric                : 1
    originate             : on
    interfaces            : []
    redistribute          : {}
    version               : 2
...
Admin@nodename# set network virtual-router test_router rip interfaces
new interface port2
Admin@nodename# show network virtual-router test_router

name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
...
rip                 :
    enabled               : off
    distance              : 120
    metric                : 1
    originate             : on
    interfaces            :
        port2
            interface           : port2
            passive-mode        : off
            poisoned-reverse    : off
            receive-version     : 0
            send-version        : 0
            split-horizone      : off

    redistribute          : {}
    version               : 2
...
```

To delete RIP router parameters, use the following command:

```
 Admin@nodename# delete network virtual-router <virtual-router-name>
rip <parameter>
```

You can delete the following parameters:

- RIP interfaces: **interfaces**.
- RIP networks: **networks-cidr**.
- Network interface from which route information updates will be sent: **networks-interface**.

Configuring multicast routing

To configure multicast routing on the virtual router, use the following command:

```
Admin@nodename# set network virtual-router <virtual-router-name>
multicast-router
```

Specify the parameters:

Parameter	Description
enabled	Enable/disable an RIP router: • **on** • **off**
ecmp	Enable traffic distribution using Equal Cost Multi Path (ECMP) technology: • **on** • **off** Requires that several routes exist to the network node of interest. If this option is disabled, all traffic to a specific destination host will be sent through a single router only (next hop).
ecmp-rebalance	

Parameter	Description
	Use ECMP rebalance: • **on**: if one of the interfaces through which traffic was sent has disconnected, then all existing flows are redistributed among the remaining routes (next hop). • **off**: if one of the interfaces through which traffic was sent has disconnected, only the flows sent through the disconnected interface will be redistributed.
join-prune	Interval for sending messages to PIM neighbors about the multicast groups whose traffic the router wants to receive or no longer wants to receive.
register-suppress	Interval after which the router sends a register suppress message.
keep-alive	Interval after which the router sends keepalive messages to neighbors, and the interval the router waits before considering a neighbor unavailable.
interfaces	Interface to use for multicasting. You can only specify interfaces added to the virtual router. To add new interfaces or change data for existing ones, use the following commands: `Admin@nodename# set network virtual-router <virtual-router-name> multicast-router interfaces new` `Admin@nodename# set network virtual-router <virtual-router-name> multicast-router interfaces <interface-name>` Parameters: • **interface**: select an interface for multicast. Only the interfaces belonging to this virtual router are available for selection. • **hello-timeout**: the interval to send PIM HELLO messages (in seconds). PIM Hello messages are sent periodically from all interfaces for which multicast support is enabled. These messages let the router know about neighbor routers that support multicasting.

Parameter	Description
	• **dr-priority**: the Designated router (DR) selection priority, which allows the administrator to control the process of DR selection for the LAN. • **bfd**: Add a bfd profile (Bidirectional Forwarding Detection). Bfd profiles are created in the element library, read more in the Configuring Libraries section. • **enable-igmp**: receive IGMP report and IGMP query messages on this interface. • **use-igmpv2**: use IGMP v2 (the default is IGMP v3).
rendezvous-points	When configuring Rendezvous points, you can specify the following parameters: • **enabled**: enable/disable this RP. ◦ **on** ◦ **off** • **name**: the RP name. • **ip**: the unicast IP address of the RP. • **asm-allowed-groups**: the list of allowed multicast group addresses for any source multicast from this RP. Any networks in the range 224.0.0.0/4. If nothing is specified, there are no restrictions.
ssm-allowed-groups	A multicast router setting that defines a list of allowed group addresses for source-specific multicast. You can specify any networks in the range 232.0.0.0/8. If nothing is specified, there are no restrictions.
spt-exclusions	A multicast router setting that defines a list of IPv4 multicast groups excluded from switching to the shortest path tree.

To display a multicast configuration of a virtual router, use the following command:

```
Admin@nodename# show network virtual-router <virtual-router-name>
multicast-router
```

Example command to configure multicast routing in a virtual router:

```
Admin@nodename# set network virtual-router test_router multicast-router
interfaces new interface port2 use-igmpv2 on
Admin@nodename# show network virtual-router test_router
```

```
name                : test_router
description         : Test virtual router
node-name           : node_1
interfaces          : port2; port3
...
multicast-router    :
    enabled               : off
    ecmp-rebalance        : off
    ecmp                  : off
    join-prune            : 60
    keep-alive            : 31
    register-suppress     : 5
    interfaces            :
        port2
            interface        : port2
            enabled          : off
            enable-igmp      : off
            use-igmpv2       : on
            bfd              : Not set

    rendevouz-points      : []
...
```

To delete multicast router parameters, use the following command:

```
 Admin@nodename# delete network virtual-router <virtual-router-name>
multicast-router
```

You can delete the following parameters:

- Interfaces used for multicast: **interfaces**.
- Rendezvous points: **rendevouz-points <rp-name>**, and the list of allowed group addresses for any source multicast from this RP: **rendevouz-points <rp-name> asm-allowed groups**.
- The list of allowed group addresses for the source-specific multicast: **ssm-allowed-groups**.

- The list of IPv4 multicast groups excluded from switching to the shortest path tree: **spt-exclusions**.

WCCP Configuration

WCCP (Web Cache Communication Protocol) settings are applied at the **network wccp** level. To create a WCCP service group, use the following command:

```
Admin@nodename# create network wccp <parameter>
```

Available parameters:

Parameter	Description
enabled	Enable/disable the service group: • **on** • **off**
name	WCCP service group name.
description	A description of the service group.
password	The password to authenticate UserGate in the service group. The password must match the one specified on the WCCP servers.
fwd-type	Forwarding type from WCCP servers to UserGate: • **l2**: use L2 redirection. In this case, the router (WCCP server) replaces a destination MAC address in the packet with a UserGate address. • **gre**: use a GRE (Generic Routing Encapsulation) tunnel. L2 redirection generally requires fewer resources than GRE, but the WCCP server and UserGate must reside in the same L2 segment. Not all WCCP server types support L2 redirection with WCCP clients.
ret-type	Forwarding type from UserGate to WCCP servers: • **l2**: using L2 redirection. In this case, UserGate (the WCCP client) changes the destination MAC address in the packet to that of the WCCP server. • **gre**: use a GRE (Generic Routing Encapsulation) tunnel.

Parameter	Description
	L2 redirection generally requires fewer resources than GRE, but the WCCP server and UserGate must reside in the same L2 segment. Not all WCCP server types support L2 redirection with WCCP clients.
service-group	The numeric ID of the service group. Service group IDs must be identical on all devices in the group.
priority	The group's priority. If multiple service groups are applicable to the traffic managed by the WCCP server, the priority determines the order in which the server will distribute traffic to the WCCP clients.
ports	Ports to redirect (traffic destination ports). If necessary, multiple ports can be specified in the **ports-to-redirect + [80 442]** format. **Important!** UserGate can only apply filtering to redirected TCP traffic with destination ports 80 and 443 (HTTP/HTTPS). Traffic sent to UserGate through other ports is sent to the Internet unfiltered.
ports-source	Redirection of traffic based on the source port values: • **on** • **off**
protocol	Select a protocol: • **tcp**: Transmission Control Protocol (TCP) • **udp**: User Datagram Protocol (UDP).
routers-lists	List of WCCP server IP addresses. For more details about how to create IP address lists using CLI, see Configuring IP Addresses.
routers-ips	WCCP server IP addresses.
assignment-type	When there are multiple WCCP clients in a service group, the assignment type determines how traffic is distributed from the WCCP servers to the WCCP clients. • **hash**: distribute traffic based on a hash computed from the specified IP packet fields. The options are: ◦ **source-ip**: calculate the hash based on the source IP address ◦ **source-port**: calculate the hash based on the source port

Parameter	Description
	◦ **dest-ip**: calculate the hash based on the destination IP address ◦ **dest-port**: calculate the hash based on the destination port ◦ **alt-source-ip**: calculate an alternate hash based on the source IP address ◦ **alt-source-port**: calculate an alternate hash based on the source port ◦ **alt-dest-ip**: calculate an alternate hash based on the destination IP address ◦ **alt-dest-port**: calculate an alternate hash based on the destination port. • **mask**: distribute traffic based on the result of a Boolean AND between the mask and the selected packet header. When selecting a mask, consult the vendor documentation for the WCCP server. ◦ **source-ip**: mask by the source IP address ◦ **source-port**: mask by the source port ◦ **dest-ip**: mask by the destination IP address ◦ **dest-port**: mask by the destination port ◦ **mask-value**: mask value for the mask scheme. 16 bits for masking by port and 32 bits for masking by IP address. Specify the value in hexadecimal format.

To specify values for a WCCP service group or update information on it, use the following command:

```
Admin@nodename# set network wccp <service-group-name> <parameter>
```

Specify the parameters to update. The parameter values are listed in the table above.

To view information about a WCCP service group:

```
Admin@nodename# show network wccp <service-group-name>
```

Example commands to create and edit WCCP:

```
Admin@nodename# create network wccp name "Test service group" protocol
tcp service-group 1 routers-ips [ 192.168.100.120 ] fwd-type l2 ret-
type l2 ports [ 80 ] priority 1 password 12345
Admin@nodename# show network wccp "Test service group"

name                : Test service group
enabled             : off
fwd-type            : l2
ret-type            : l2
service-group       : 1
priority            : 1
protocol            : tcp
ports               : 80
assignment-type     : hash
source-ip           : off
source-port         : off
dest-ip             : off
dest-port           : off
alt-source-ip       : off
alt-source-port     : off
alt-dest-ip         : off
alt-dest-port       : off
routers-ips         : 192.168.100.120
Admin@nodename# set network wccp "Test service group" description "Test
service group description" service-group 100
Admin@nodename# show network wccp "Test service group"

name                : Test service group
description         : Test service group description
enabled             : off
fwd-type            : l2
ret-type            : l2
service-group       : 100
priority            : 1
protocol            : tcp
ports               : 80
assignment-type     : hash
source-ip           : off
source-port         : off
```

```
dest-ip             : off
dest-port           : off
alt-source-ip       : off
alt-source-port     : off
alt-dest-ip         : off
alt-dest-port       : off
routers-ips         : 192.168.100.120
```

To remove a service group completely or some of its parameters:

```
Admin@nodename# delete network wccp <service-group-name>
```

You can delete the following parameters:

- **routers-lists**.
- **routers-ips**.
- **ports**.

CONFIGURING THE USERS AND DEVICES SECTION

Configuring User Groups

You configure user groups at the **users group** levels.

To add a new user group, use the following command:

```
Admin@nodename# create users group <parameter>
```

You can specify the following parameters:

Parameter	Description
name	The user group name.

Parameter	Description
description	The user group description.
transient	Values: • **on**: the group is for guest users • **off**: the group is not for guest users.
users	Add users to the group.
ldap-users	Add LDAP users. When adding LDAP users, specify an LDAP connector (**ldap-users connector <ldap-server-name> users + [<domain\user1> <domain\user2> ...]**).

To edit information about a user group, use the following command (the parameters available to update are the same as those for creating a group):

```
Admin@nodename# set users group <group-name> <parameter>
```

To display group settings, use the following command:

```
Admin@nodename# show users group <group-name>
```

Example commands to create and edit a user group:

```
Admin@nodename# create users group name "Test user group" ldap-users
connector "LDAP connector" users [ testd.local\user1 ]
Admin@nodename# show users group "Test user group"

name            : Test user group
is-ldap         : off
is-transient    : off
users           : user1 user1 (testd.local\user1)
Admin@nodename# set users group "Test user group" users [ user2 ]
Admin@nodename# show users group "Test user group"

name            : Test user group
is-ldap         : off
```

```
is-transient    : off
users           : user2; user1 user1 (testd.local\user1)
```

To delete a user group or individual users from it, use the following commands:

```
Admin@nodename# delete users group <group-name>
```

To delete local users, use the following command:

```
Admin@nodename# delete users group <group-name> users [ <user1>
<user2> ... ]
```

To delete LDAP users, use the following command:

```
Admin@nodename# delete users group <group-name> ldap-users connector
<ldap-server-name> users [ <domain\user1> <domain\user2> ... ]
```

Example of removing an LDAP user from a group:

```
Admin@nodename# delete users group "Test user group" ldap-users
connector "LDAP connector" users [ testd.local\user1 ]
```

Configuring Users

You configure users at the **users user** level.

To add users, use the following command:

```
Admin@nodename# create users user <parameter>
```

Available parameters:

Parameter	Description
enabled	Enable/disable the user.

Parameter	Description
name	The username.
login	User login for login/password identification. In this case, you will need to configure the captive portal where a user can enter their login name and password for authentication.
password	User password for login/password identification. In this case, you will need to configure the captive portal where a user can enter their login name and password for authentication.
expiration-date	The expiration date for the user account. Format: YYYY-MM-DD.
groups	Groups to add the user to.
ip	IP addresses to identify the user. The user must always access the network from the specified addresses.
mac	MAC addresses to identify the user. The user must always access the network from the specified addresses.
ip-range	The IP address range to identify the user. The user must always access the network from an address in the specified range. Format: <IP_start-IP_end>.
ip-mac	A combination of MAC and IP addresses to identify the user. The user must always access the network from the specified addresses. Format: <ip-mac>.
vlan-tag	The VLAN tag for user identification.
emails	User's email addresses.
phones	User's phone numbers.

To update parameters of a user account, use the following command:

```
Admin@nodename# set users user <user-name> <parameter>
```

The list of available parameters is the same as the list for creating a user account.

To view a user account, use the following command:

```
Admin@nodename# show users user <user-name>
```

Example commands to create and edit a user account:

```
Admin@nodename# create users user name user_2 login user2 password
12345 expiration-date 2023-12-31 ip [ 192.168.100.112 ] enabled on
Admin@nodename# show users user user_2

name               : user_2
login              : user2
enabled            : on
expiration-date    : December 31, 2023, 00:00
ip                 : 192.168.100.112
Admin@nodename# set users user user_2 emails [ example@example.org ]
Admin@nodename# show users user user_2

name               : user_2
login              : user2
enabled            : on
emails             : example@example.org
expiration-date    : December 31, 2023, 00:00
ip                 : 192.168.100.112
```

To delete a user account, use the following command:

```
Admin@nodename# delete users user <user-login>
```

You can also delete specific data from a user account. Available parameters are (you must specify parameter values to delete):

- **groups**
- **static-addresses**
- **emails**
- **phones**

Configuring Authentication Servers

The **Auth servers** section allows you to configure an LDAP connector, RADIUS, TACACS+, NTLM, and SAML IDP servers. You configure auth servers at the **users auth-server** level. We will consider it in the respective sections below.

Configuring LDAP connectors

An LDAP connector is configured at the **users auth-servers ldap** level.

To create an LDAP connector, use the following command:

```
Admin@nodename# create users auth-server ldap <parameter>
```

Provide the following parameters:

Parameter	Description
name	LDAP connector name.
enabled	Enable/disable the auth server.
description	LDAP connector description.
ssl	Values: • **on**: use an SSL connection to connect to the LDAP server • **off**: connect to the LDAP server without using an SSL connection.
address	Controller IP address or the LDAP domain name.
bind-dn	The username used to connect to the server. Format: DOMAIN\username or username@domain. The user must be a user in the domain.
password	The user's password for connecting to the domain.
cache-ttl	LDAP cache entry lifetime. (This option is available starting from UGOS 7.1.3).
domains	List of domains served by the domain controller.
search-roots	The list of LDAP server paths relative to which the system will search for users and groups. Specify the full name, e.g.,

Parameter	Description
	ou=Office,dc=example,dc=com. If the search paths are not specified, the system will search over the entire directory, starting from the root.

To edit information about an existing LDAP connector, use the following command:

```
 Admin@nodename# set users auth-server ldap <ldap-server-name>
<parameter>
```

The parameters available to update are the same as those for creating an LDAP connector.

To display information on an LDAP connector, use the following command:

```
Admin@nodename# show users auth-server ldap <ldap-server-name>
```

Example commands to create and edit an LDAP connector:

```
Admin@nodename# create users auth-server ldap name "New LDAP connector"
ssl on address 10.10.0.10 bind-dn ug@testd.local password 12345 domains
[ testd.local ] search-roots [ dc=testd,dc=local ] enabled on
Admin@nodename# show users auth-server ldap "New LDAP connector"

name              : New LDAP connector
enabled           : on
ssl               : on
address           : 10.10.0.10
bind-dn           : ug@testd.local
domains           : testd.local
search-roots      : dc=testd,dc=local
keytab_exists     : off
Admin@nodename# set users auth-server ldap "New LDAP connector"
description "New LDAP connector description"
Admin@nodename# show users auth-server ldap "New LDAP connector"

name              : New LDAP connector
description       : New LDAP connector description
```

```
enabled          : on
ssl              : on
address          : 10.10.0.10
bind-dn          : ug@testd.local
domains          : testd.local
search-roots     : dc=testd,dc=local
keytab_exists    : off
```

To delete an LDAP connector, use the following command:

```
 Admin@nodename# delete users auth-server ldap <ldap-server-name>
<parameter>
```

You can also delete individual parameters of an LDAP connector. You can delete the following parameters:

- **domains**
- **search-roots**

Configuring RADIUS Servers

A RADIUS server is configured at the **users auth-servers radius** level.

To create a RADIUS auth server, use the following command:

```
Admin@nodename# create users auth-server radius <parameter>
```

Provide the following parameters:

Parameter	Description
name	The RADIUS server name.
enabled	Enable/disable the auth server.
description	Auth server description.
secret	Pre-shared key used by the RADIUS protocol for authentication.
addresses	

Parameter	Description
	IP address and the UDP port on which the RADIUS server listens to requests (default port: 1812). Format: <ip:port>.

To update information about a RADIUS server, use the following command:

```
 Admin@nodename# set users auth-server radius <radius-server-name>
<parameter>
```

The parameters you can update are the same as those used to create an auth server.

To display information about a RADIUS server, use the following command:

```
Admin@nodename# show users auth-server radius <radius-server-name>
```

Example commands to create and edit a RADIUS server:

```
Admin@nodename# create users auth-server radius name "New RADIUS
server" addresses [ 10.10.0.9:1812 ] secret 12345 enabled on
Admin@nodename# show users auth-server radius "New RADIUS server"

name           : New RADIUS server
enabled        : on
addresses      :
    host    : 10.10.0.9
    port    : 1812
Admin@nodename# set users auth-server radius "New RADIUS server"
description "New RADIUS server description"
Admin@nodename# show users auth-server radius "New RADIUS server"

name           : New RADIUS server
description    : New RADIUS server description
enabled        : on
addresses      :
    host    : 10.10.0.9
    port    : 1812
```

To delete a server, use the following command:

```
 Admin@nodename# delete users auth-server radius <radius-server-name>
<parameter>
```

You can also delete individual parameters of a RADIUS server. You can delete the following parameters:

- **addresses**

Configuring a TACACS+ server

A TACACS+ server is configured at the **users auth-servers tacacs** level.

To create a TACACS+ auth server, use the following command:

```
Admin@nodename# create users auth-server tacacs <parameter>
```

Provide the following parameters:

Parameter	Description
name	TACACS+ server name.
enabled	Enable/disable the server.
description	Auth server description.
secret	Pre-shared key used by the TACACS+ protocol for authentication.
address	The IP address for the TACACS+ server.
port	The UDP port on which the TACACS+ server listens for authentication requests. By default, UDP port 1812 is used.
single-connection	Use a single TCP connection for communicating with the TACACS+ server.
timeout	The authentication timeout for the TACACS+ server. The default is 4 seconds.

To edit information about a TACACS+ server, use the following command:

```
Admin@nodename# set users auth-server tacacs <tacacs-server-name>
<parameter>
```

The parameters you can update are the same as those used to create an auth server.

To display information about a TACACS+ server, use the following command:

```
Admin@nodename# show users auth-server tacacs <tacacs-server-name>
```

Example commands to create and edit a TACACS+ server:

```
Admin@nodename# create users auth-server tacacs address 10.10.0.11 name
"New TACACS+ server" port 1812 secret 12345 enabled on
Admin@nodename# show users auth-server tacacs "New TACACS+ server"

name                 : New TACACS+ server
enabled              : on
address              : 10.10.0.11
port                 : 1812
single-connection    : off
timeout              : 4
Admin@nodename# set users auth-server tacacs "New TACACS+ server"
description "New TACACS+ server description"
Admin@nodename# show users auth-server tacacs "New TACACS+ server"

name                 : New TACACS+ server
description          : New TACACS+ server description
enabled              : on
address              : 10.10.0.11
port                 : 1812
single-connection    : off
timeout              : 4
```

To delete a server, use the following command:

```
 Admin@nodename# delete users auth-server tacacs <tacacs-server-name>
```

Configuring NTLM servers

An NTLM server is configured at the **users auth-servers ntlm** level.

To create an NTLM auth server, use the following command:

```
Admin@nodename# create users auth-server ntlm <parameter>
```

Provide the following parameters:

Parameter	Description
name	The NTLM server name.
enabled	Enable/disable the auth server.
description	Auth server description.
domain	The IP address or domain name of the NLM server.

To update information about an NTLM server, use the following command:

```
 Admin@nodename# set users auth-server ntlm <ntlm-server-name>
<parameter>
```

To display information about an NTLM server, use the following command:

```
Admin@nodename# show users auth-server ntlm <ntlm-server-name>
```

The parameters you can update are the same as those used to create an auth server.

Example commands to create and edit an NTLM server:

```
Admin@nodename# create users auth-server ntlm name "New NTLM server"
domain 10.10.0.12 enabled on
Admin@nodename# show users auth-server ntlm "New NTLM server"

name            : New NTLM server
enabled         : on
```

```
domain          : 10.10.0.12

Admin@nodename# set users auth-server ntlm "New NTLM server"
description "New NTLM server description"
Admin@nodename# show users auth-server ntlm "New NTLM server"

name            : New NTLM server
description     : New NTLM server description
enabled         : on
domain          : 10.10.0.12
```

To delete a server, use the following command:

```
Admin@nodename# delete users auth-servers ntlm <ntlm-server-name>
```

Configuring an SAML IDP server

A SAML IDP server is configured at the **users auth-servers saml-idp** level.

To create an SAML IDP auth server, use the following command:

```
Admin@nodename# create users auth-server saml-idp <parameter>
```

Provide the following parameters:

Parameter	Description
name	SAML IDP server name.
enabled	Enable/disable the auth server.
description	Auth server description.
metadata-url	The URL on the SAML IDP server from where an XML file with a valid configuration for this SAML service provider (client) can be downloaded.
certificate	The certificate that will be used on the SAML client.
sso-url	The URL that is used on the SAML IDP server as the single login point. For more details, see the documentation for your SAML IDP server.

Parameter	Description
sso-binding	The method used to work with a SSO single login point. Options: POST and Redirect. For more details, see the documentation for your SAML IDP server.
slo-url	The URL used on the SAML IDP server as the single logout point. For more details, see the documentation for your SAML IDP server.
slo-binding	The method used to work with a SSO single logout point. Options: POST and Redirect. For more details, see the documentation for your SAML IDP server.

To update information about a SAML IDP server, use the following command:

```
 Admin@nodename# set users auth-server saml-idp <saml-idp-server-name>
<parameter>
```

The parameters you can update are the same as those used to create an auth server.

To display information about a SAML IDP server, use the following command:

```
Admin@nodename# show users auth-server saml-idp <saml-idp-server-name>
```

Example commands to create and edit a SAML IDP server:

```
Admin@nodename# create users auth-server saml-idp name "New SAML IDP
server" slo-url http://logout.example.org sso-url http://
login.example.o
rg enabled on
Admin@nodename# show users auth-server saml-idp "New SAML IDP server"

name            : New SAML IDP server
enabled         : on
certificate     : Unused
sso-url         : http://login.example.org
sso-binding     : post
slo-url         : http://logout.example.org
slo-binding     : post
```

```
Admin@nodename# set users auth-server saml-idp "New SAML IDP server"
description "New SAML IDP server description"
Admin@nodename# show users auth-server saml-idp "New SAML IDP server"

name            : New SAML IDP server
description     : New SAML IDP server description
enabled         : on
certificate     : Unused
sso-url         : http://login.example.org
sso-binding     : post
slo-url         : http://logout.example.org
slo-binding     : post
```

To delete a server, use the following command:

```
 Admin@nodename# delete users auth-servers saml-idp <saml-idp-server-
name>
```

Configuring Authentication Profiles

You configure auth profiles at the **users auth-profile** level.

To create an auth profile, use the following command:

```
Admin@nodename# create users auth-profile <parameter>
```

Provide the following parameters:

Parameter	Description
name	The name of the MFA profile.
description	A description of the MFA profile.
mfa	Specify the multifactor authentication profile (if it is required). An MFA profile you specify must be already created. For more details about creating MFA profiles using CLI, see Configuring MFA (Multifactor Authentication) Profiles.

Parameter	Description
idle-time	Idle time before disconnection (in seconds). After the specified time without activity the user's status will change to **Unknown user**.
expiration-time	Authorized user time-to-live (in seconds). After the specified time the user's status will change to **Unknown user** requiring the user to authorize on the captive portal again.
max-attempts	Max authorization failures through the Captive portal allowed before the user account is locked.
lockout-time	Time (in seconds) for which the user account is locked if the specified number of max failures is reached.
auth-methods	Authentication method: • **local-user-auth**: authentication using the local user database. • **policy-accept**: no authentication is required, but the user must agree to the network usage policy before accessing the Internet. This is used with the Captive portal profile which uses the Captive portal policy authorization page. • **http-basic**: authentication using the HTTP Basic method. • **ldap**: authentication using an LDAP connector. • **radius**: authentication using a RADIUS server. • **tacacs**: authentication using a TACACS+ server. • **ntlm**: authentication using an NTLM server. • **saml-idp**: authentication using an SAML IDP server.

To edit authentication profile parameters, use the following command:

```
Admin@nodename# set users auth-profile <auth-profile-name> <parameter>
```

The list of parameters available to update is the same as for the **create** command.

Example of creating and editing a user authentication profile:

```
Admin@nodename# create users auth-profile name "New LDAP auth profile"
auth-methods ldap [ "New LDAP connector" ]
Admin@nodename# show users auth-profile "New LDAP auth profile"
```

```
name                : New LDAP auth profile
max-attempts        : 5
idle-time           : 900
expiration-time     : 86400
lockout-time        : 300
mfa                 : none
auth-methods        :
    http-basic          : off
    local-user-auth     : off
    policy-accept       : off
    ldap                : New LDAP connector
Admin@nodename# set users auth-profile "New LDAP auth profile"
description "New LDAP auth profile description"
Admin@nodename# show users auth-profile "New LDAP auth profile"

name                : New LDAP auth profile
description         : New LDAP auth profile description
max-attempts        : 5
idle-time           : 900
expiration-time     : 86400
lockout-time        : 300
mfa                 : none
auth-methods        :
    http-basic          : off
    local-user-auth     : off
    policy-accept       : off
    ldap                : New LDAP connector
```

You can use the command line interface to delete an entire profile or individual authentication methods specified in a profile. To do this, use the following commands.

To delete an authentication profile:

```
Admin@nodename# delete users auth-profile <auth-profile-name>
```

To delete authentication methods configured in a profile, you need to specify an authentication method (available authorization methods are listed in the table above):

```
Admin@nodename# delete users auth-profile <auth-profile-name> auth-
methods <auth-metod>
```

Configuring Captive Profiles

Captive profiles are configured at the **users captive-profiles** level.

To create a Captive profile, use the following command:

```
Admin@nodename# create users captive-profiles <parameter>
```

Provide the following parameters:

Parameter	Description
name	Captive profile name.
description	Captive profile description.
auth-template	Auth template.
auth-mode	Authentication mode UserGate uses to "remember" a user: • **ip**: use the IP address. After a user successfully authenticates through the Captive portal, UserGate remembers the user's IP address, and any subsequent connection from that IP address will be attributed to that user. This is the default method. • **cookie**: store the cookie. After a user successfully authenticates through the Captive portal, UserGate adds a cookie to the user's browser to identify subsequent connections by that user.
auth-profile	Authentication profile that defines authentication methods. For more details on configuring authentication profiles using the CLI, see the Configuring Authentication Profiles section.
custom-redirect	URL to redirect the user to after successful authentication using the Captive portal. If not specified, the user is redirected to the URL they requested.
use-cookie	

Parameter	Description
	Option to save authentication in the browser for a specified time interval. This information is saved in a cookie. • **on** • **off**
cookie-exptime	Time for which authentication is saved (in hours).
enable-ldap	Option to choose an AD/LDAP domain on the login page: • **on** • **off**
use-captcha	Prompt a user for a code shown on the Captive portal login page: • **on** • **off**
use-https	Use HTTPS when displaying the Captive portal authentication page. A properly configured captive portal SSL certificate is required. • **on** • **off**
notification-profile	The notification profile for sending information about the created user and their password to guest users. For more details on configuring notification profiles using the CLI, see Configuring Notification Profiles.
notification-sender	Sender of the notification. Specify a name (if using an SMPP profile) or an email (if using an SMTP profile).
notification-subject	Subject of the notification, if using email notifications.
notification-body	Body of the email. In the message body, you can use special variables named {login} and {password} that will be replaced with the username and password, respectively. The notification text is separated by quotation marks ("").
exp-time	Date and time to disable a temporary user account. The required format is: *yyyy-mm-ddThh:mm:ssZ*.
session-ttl	Amount of time (in hours) from the first temporary user authentication, after which their account will be disabled.

Parameter	Description
password-len	The password length is 1 to 15 characters.
password-complexity	Password complexity: • **num**: numbers only. • **alpha_num**: numbers and letters. • **alpha_num_special**: numbers, letters, and special characters.
ta-groups	The groups to which the created guest users will be added.
captive-auth-mode	Select Captive profile authentication method: • aaa: authenticate using a local user login/password or an AAA server. • pki: X.5098 certificate-based authentication.
uc-profile	Select the user certificate profile for PKI-based authentication.

To edit a profile, use the following command:

```
Admin@nodename# set users captive-profiles <captive-profile-name>
<parameter>
```

The parameters available to update for a captive profile are the same as those for creating a profile.

To display captive profile settings, use the following command:

```
Admin@nodename# show users captive-profiles <captive-profile-name>
```

Example of creating and editing a captive profile:

```
Admin@nodename# create users captive-profiles name "New captive
profile" auth-profile "LDAP auth profile" captive-auth-mode aaa enable-
ldap on
Admin@nodename# set users captive-profiles "New captive profile" use-
https on
```

To delete a profile, use the following command:

```
Admin@nodename# delete users captive-profiles <captive-profile-name>
```

To delete a temporary user group (you need to have at least one temporary user group specified), use the following command:

```
Admin@nodename# delete users captive-profiles <captive-profile-name>
ta-groups
```

Captive portal

This section describes how to configure Captive portal rules. You configure them at the **users captive-portal** level. For more details on the command structure, see UserGate Policy Language.

Captive portal rule parameters:

Parameter	Description
OK **PASS**	Actions for a Captive portal rule: • **OK**: use authentication • **PASS**: do not use authentication
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	The name of the captive portal rule. Example: **name("Captive rule example")**.
desc	A description of the captive portal rule. Example rule specification: **desc("Captive portal rule example set via CLI")**.

Parameter	Description
profile	When using the Captive portal authentication, specify the Captive profile. Example: **profile("Example Captive profile")**. For more details about creating and configuring Captive profiles, see Configuring Captive Profiles.
rule_log	Enable/disable logging when a rule was triggered: • **rule_log(yes)** or **rule_log(true)** • **rule_log(no)** or **rule_log(false)** If this parameter is not specified, logging is disabled.
src.zone	Source zone To specify a source zone, such as Trusted: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add source IP address or domain lists. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
dst.zone	Traffic destination zone. To specify a destination zone, such as Untrusted: **dst.zone = Untrusted**. For more details about how to configure zones using CLI, see the Zones section.
dst.ip	Add lists of destination IP addresses or domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in

Parameter	Description
	parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
dst.geoip	To specify a destination GeoIP, it is necessary to specify a country code (for example, **dst.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
url	The URL lists to which the rule will be applied. To specify a URL list: **url = lib.url()**. Specify a URL list name in parentheses.
time	Set a schedule for a rule. To set a schedule: **time = lib.time()**. Specify a time set group name in parentheses. For more details on configuring time sets, see Configuring time sets.

Example of creating and adding a captive portal rule using UPL:

```
Admin@nodename# create users captive-portal 1 upl-rule OK \
...profile("New captive profile") \
...rule_log(true) \
...name("Captive portal rule new") \
...
Admin@nodename# show users captive-portal 1
% ----------------- 1 -----------------
OK \
    rule_log(yes) \
    profile("New captive profile") \
    enabled(false) \
    id("676df2b1-03e9-42b2-8375-0b8f78c4c47c") \
    name("Captive portal rule new")

Admin@nodename# set users captive-portal 1 upl-rule OK \
...src.zone = Trusted \
...dst.zone = Untrusted
...
Admin@nodename# show users captive-portal 1
```

```
% ---------------- 1 ----------------
OK \
    src.zone = Trusted \
    dst.zone = Untrusted \
    rule_log(yes) \
    profile("New captive profile") \
    enabled(false) \
    id("676df2b1-03e9-42b2-8375-0b8f78c4c47c") \
    name("Captive portal rule new")
```

Configuring Terminal Servers

This section describes how to configure terminal servers using the CLI. You configure them at the **users terminal-servers** level.

To create a terminal server, use the following command:

```
Admin@nodename# create users terminal-servers <parameter>
```

Provide the following parameters:

Parameter	Description
enabled	Enable/disable a terminal server: • **on** • **off**
name	Terminal server name.
description	Terminal server description.
hosts	Host IP address. To add more than one address, separate them with commas.

To edit a terminal server's parameters (listed in the table above), use the following command:

```
Admin@nodename# set users terminal-servers <terminal-server-name>
<parameter>
```

To display information about a terminal server, use the following command:

```
Admin@nodename# show users terminal-servers <terminal-server-name>
```

Example of creating and editing a terminal server:

```
Admin@nodename# create users terminal-servers name "Test terminal
server" hosts [ 10.10.0.20 ] enabled on
Admin@nodename# show users terminal-servers "Test terminal server"

name           : Test terminal server
enabled        : on
hosts          : 10.10.0.20

Admin@nodename# set users terminal-servers "Test terminal server"
description "Test terminal server description"
Admin@nodename# show users terminal-servers "Test terminal server"

name           : Test terminal server
description    : Test terminal server description
enabled        : on
hosts          : 10.10.0.20
```

To delete a terminal server, use the following command:

```
Admin@nodename# delete users terminal-servers <terminal-server-name>
```

You can also delete individual hosts. To delete them specify their addresses:

```
Admin@nodename# delete users terminal-servers <terminal-server-name>
hosts
```

Configuring MFA (Multifactor Authentication) Profiles

This section describes how to configure multifactor authentication profiles using CLI. You configure MFA profiles at the **users mfa-profiles** level. You can create multiple types of profiles:

- **MFA by TOTP**: use a Time-based One Time Password (TOTP) token as the second authentication factor.
- **MFA by email**: use a one-time password received by email as the second authentication factor.
- **MFA by SMS**: use a one-time password received by SMS as the second authentication factor.

To create a multifactor authentication profile, use the following command:

```
Admin@nodename# create users mfa-profiles <parameter>
```

To delete a multifactor authentication profile, use the following command:

```
Admin@nodename# delete users mfa-profiles <mfa-name>
```

To display information about all or individual MFA profiles, use the following commands:

```
Admin@nodename# show users mfa-profiles
Admin@nodename# show users mfa-profiles <mfa-name>
```

Configuring MFA by TOTP

To add a new profile for multifactor authentication via TOTP, use the following command:

```
Admin@nodename# create users mfa-profiles totp <parameter>
```

Provide the following parameters:

Parameter	Description
name	The name of the MFA profile.
description	A description of the MFA profile.
show-qr-code	QR code on the Captive portal page or in an email to facilitate configuring the device or the TOTP client software.
notification-profile	Select the notification profile to use.
notification-sender	Sender of the notification. Specify a name (if using an SMPP profile) or an email (if using an SMTP profile).
notification-subject	Subject of the notification, if using email notifications.
notification-body	Body of the email. In the message body, you can use a special variable named {2fa_auth_code} that will be replaced by the one-time password. The notification text is separated by quotation marks ("").

To edit a profile for multifactor authentication via TOTP, use the following command:

```
Admin@nodename# set users mfa-profiles totp <mfa-totp-name> <parameter>
```

The parameters available to edit are identical to those used to create a profile.

Example of creating and editing a profile for multifactor authentication via TOTP:

```
Admin@nodename# create users mfa-profiles totp name "Test TOTP MFA
profile" notification-profile pass show-qr-code on
Admin@nodename# show users mfa-profiles totp "Test TOTP MFA profile"

name                    : Test TOTP MFA profile
show-qr-code            : on
notification-profile    : pass
notification-body       : Your authentication code is {2fa_auth_code}!
Do not share it with anybody!
Admin@nodename# set users mfa-profiles totp "Test TOTP MFA profile"
description "Test TOTP MFA profile description"
Admin@nodename# show users mfa-profiles totp "Test TOTP MFA profile"
```

```
name                    : Test TOTP MFA profile
description             : Test TOTP MFA profile description
show-qr-code            : on
notification-profile    : pass
notification-body       : Your authentication code is {2fa_auth_code}!
Do not share it with anybody!
```

Configuring MFA by email

To add a new profile for multifactor authentication via email, use the following command:

```
Admin@nodename# create users mfa-profiles smtp <parameter>
```

Provide the following parameters:

Parameter	Description
name	The name of the MFA profile.
description	A description of the MFA profile.
notification-profile	Select the notification profile to use.
notification-sender	Email of the notification sender.
notification-subject	Notification subject.
notification-body	Body of the email. In the message body, you can use a special variable named {2fa_auth_code} that will be replaced by the one-time password. The notification text is separated by quotation marks ("").
code-lifetime	One-time password validity period (in seconds).

To edit a profile for multifactor authentication via email, use the following command:

```
 Admin@nodename# set users mfa-profiles smtp <mfa-email-profile>
<parameter>
```

The parameters available to update are identical to those used to create a profile.

Example of creating and editing a profile for multifactor authentication via email:

```
Admin@nodename# create users mfa-profiles smtp name "Test SMTP MFA
profile" notification-profile "Example SMTP profile" notification-
sender sender@example.org notification-subject "Test notification subj"
notification-body "Test notification text"
Admin@nodename# show users mfa-profiles smtp "Test SMTP MFA profile"

name                    : Test SMTP MFA profile
notification-profile    : Example SMTP profile
notification-sender     : sender@example.org
notification-subject    : Test notification subj
notification-body       : Test notification text
code-lifetime           : 60
Admin@nodename# set users mfa-profiles smtp "Test SMTP MFA profile"
code-lifetime 70
Admin@nodename# show users mfa-profiles smtp "Test SMTP MFA profile"

name                    : Test SMTP MFA profile
notification-profile    : Example SMTP profile
notification-sender     : sender@example.org
notification-subject    : Test notification subj
notification-body       : Test notification text
code-lifetime           : 70
```

Configuring MFA by SMS

To add a new profile for multifactor authentication via SMS, use the following command:

```
Admin@nodename# create users mfa-profiles smpp <parameter>
```

Provide the following parameters:

Parameter	Description
name	The name of the MFA profile.
description	A description of the MFA profile.

Parameter	Description
notification-sender	Name of the notification sender.
notification-body	Body of the email. In the message body, you can use a special variable named {2fa_auth_code} that will be replaced by the one-time password. The notification text is separated by quotation marks ("").
code-lifetime	One-time password validity period (in seconds).

To edit a profile for multifactor authentication via SMS, use the following command:

```
 Admin@nodename# set users mfa-profiles smpp <mfa-sms-profile>
<parameter>
```

The parameters available to update are identical to those used to create a profile.

Example of creating and editing a profile for multifactor authentication via SMS:

```
Admin@nodename# create users mfa-profiles smpp name "Test SMPP MFA
profile" notification-profile "Example SMPP profile" notification-
sender Tes_sender notification-body "Test notification text"
Admin@nodename# show users mfa-profiles smpp "Test SMPP MFA profile"

name                    : Test SMPP MFA profile
notification-profile    : Example SMPP profile
notification-sender     : Tes_sender
notification-body       : Test notification text
code-lifetime           : 60
Admin@nodename# set users mfa-profiles smpp "Test SMPP MFA profile"
code-lifetime 80
Admin@nodename# show users mfa-profiles smpp "Test SMPP MFA profile"

name                    : Test SMPP MFA profile
notification-profile    : Example SMPP profile
notification-sender     : Tes_sender
notification-body       : Test notification text
code-lifetime           : 80
```

Viewing Information About Authorized Users

To view information about authorized users, use the following command line interface command in monitoring mode:

```
Admin@nodename> show user-auth
```

To view the details of an authentication session for a specific user, use the command:

```
Admin@nodename> show user-auth <user name>
```

To delete a session of a specific user, use the command:

```
Admin@nodename> clear user-auth <parameter>
```

where either the username or the IP address can be used as the parameter.

Configuring Policy Application to Users

For the local UserGate device users the policies are applied automatically.

If users authenticate via an LDAP connector, NTLM, or Kerberos, then to apply policies to users (in cases of adding a new LDAP group or user to a group, creating a rule, and applying it to an LDAP group), it is necessary to reset the sessions of all users and clear the LDAP record cache on UserGate.

You can reset sessions of individual users using the CLI. The command is executed in configuration mode (configure), to execute the command you need to know the user's IP address:

```
Admin@nodename# execute termination user-sessions ip <IP-address>
```

To clear the cache, use the command:

```
Admin@nodename# execute cache ldap-clear
```

Configuring UserID Agent

The User'ID agent is designed to perform transparent authentication on selected UserGate devices. Microsoft Active Directory logs (via the WMI protocol), syslog (via the standardized syslog protocol RFC 3164, RFC 5424, RFC 6587), and RADIUS logs (starting from software version 7.2.0) are used as the source of the authentication data. The detailed information on the UserID agent can be found in the Users and devices section of the DCFW Administrator Guide.

You configure UserID in the CLI at the **users userid-agent** level.

Configuring the UserID agent settings

The general settings of a UserID agent are configured using the following command:

```
Admin@nodename# set users userid-agent configurate-agent <parameters>
```

To configure, you need to specify the following parameters:

Parameter	Description
polling-interval	Active Directory servers polling interval. The default value is 120 seconds.
syslog-monitoring-interval	Database poll period to look for syslog-source user session start/end events.
radius-monitoring-interval	Database poll period to look for user session start/end events in the RADIUS log. (This option is available starting from software version 7.2.0 and up).
ignore-network-list	Lists of IP addresses the events from which should be ignored by the UserID agent. A record about the ignored source appears in the **UserID agent** log. The list can be created in the libraries**(IP addresses)** section. This setting is global and applies to all sources.
ignore-user-list	Names of users the events from which should be ignored by the UserID agent. The search is based on the Common Name (CN) of the AD user.

Parameter	Description
	This setting is global and applies to all sources. An entry about an ignored user will appear in the **UserID** log. **Important!** When specifying a name, you can use the asterisk (*), but only at the end of a string.
tcp-enabled	The TCP protocol for collecting logs using the Syslog protocol: • **on** • **off**
udp-enabled	The UDP protocol for collecting logs using the Syslog protocol: • **on** • **off**

Configuring Event Source

Microsoft Active Directory

To add Microsoft Active Directory as an event source, use the following command:

```
Admin@nodename# create users userid-agent active-directory <parameters>
```

To configure, you need to specify the following parameters:

Parameter	Description
enabled	Enable/disable receiving logs from the source.
name	The source name.
description	An optional description of the source.
address	Microsoft Active Directory address.
protocol	AD access protocol (WMI).
login	The username for connecting to AD.
password	The user's password for connecting to AD.
auth-profile	The authentication profile used to look up users found in AD logs.

Parameter	Description
expiration-time	The period of time after which the user's session will be forcibly terminated. The default value is 2700 seconds (45 minutes).

To edit a previously created Active Directory event source, use the following command:

```
Admin@nodename# set users userid-agent active-directory <source-name>
<parameters>
```

The parameters for editing are similar to the parameters used when creating an event source of the Microsoft Active Directory type in the table above.

To view the parameters of previously created Active Directory event sources, use the command:

```
Admin@nodename# show users userid-agent active-directory
Admin@nodename# show users userid-agent active-directory <source-name>
```

To delete a previously created Active Directory event source, use the following command:

```
Admin@nodename# delete users userid-agent active-directory <source-
name>
```

Syslog-sender

To add a syslog sender as an event source, use the following command:

```
Admin@nodename# create users userid-agent syslog-sender <parameters>
```

To configure, you need to specify the following parameters:

Parameter	Description
enabled	Enable/disable receiving logs from the source.
name	The source name.

Parameter	Description
description	The source description.
address	The host address from which UserGate will receive syslog events.
default-domain	The name of the domain used to search for users found in syslog logs.
timezone	The time zone set on the source.
filters	Filters to find the necessary log entries. You can create and configure filters under **Libraries → UserID agent syslog filters of the agent.** For more details, see UserID agent Syslog filters.
auth-profile	The authentication profile used to search for users found in **sys log** logs.
expiration-time	The period of time after which the user's session will be forcibly terminated. The default value is 2700 seconds (45 minutes).

To edit a previously created syslog sender event source, use the following command:

```
Admin@nodename# set users userid-agent syslog-sender <source-name>
<parameters>
```

The parameters for editing are similar to the parameters used when creating an event source of the syslog sender type in the table above.

To view the parameters of previously created syslog sender event sources, use the command:

```
Admin@nodename# show users userid-agent syslog-sender
Admin@nodename# show users userid-agent syslog-sender <source-name>
```

To delete a previously created syslog sender event source, use the following command:

```
Admin@nodename# delete users userid-agent syslog-sender <source-name>
```

RADIUS server

This option is available starting from software version 7.2.0 and up.

To add a RADIUS server as an event source, use the following command:

```
Admin@nodename# create users userid-agent radius-server <parameters>
```

To configure, you need to specify the following parameters:

Parameter	Description
enabled	Enable/disable receiving logs from the source.
name	The source name.
description	The source description.
address	The host addresses from which UserGate will receive events via the RADIUS protocol.
server-secret	A pre-shared key used by the RADIUS protocol for authentication.
default-domain	The name of a domain in which a user will be searched for in case the request does not indicate which domain they belong to.
attribute-for-group	The radius attribute type number in which the user's group resides, by default the group is not checked.
attribute-for-name	The radius attribute type number in which the username resides, 1 by default.
auth-profile	The authentication profile used to look up users found in RADIUS logs.
expiration-time	The period of time after which the user's session will be forcibly terminated. The default value is 2700 seconds (45 minutes).

To edit the previously created RADIUS event source, use the following command:

```
Admin@nodename# set users userid-agent radius-server <source-name>
<parameters>
```

The parameters for editing are similar to the parameters used when creating an event source of the RADIUS type in the table above.

To view the parameters of previously created RADIUS event sources, use the command:

```
Admin@nodename# show users userid-agent radius-server
Admin@nodename# show users userid-agent radius-server <source-name>
```

To delete a previously created RADIUS event source, use the following command:

```
Admin@nodename# delete users userid-agent radius-server <source-name>
```

CONFIGURING THE NETWORK POLICIES SECTION

Configuring Firewall Rules

You configure a firewall at the **network-policy firewall** level. For more details on the command structure, see UserGate Policy Language.

```
Admin@nodename# create network-policy firewall
```

Firewall rule parameters:

Parameter	Description
PASS **DENY**	Firewall rule action: • **PASS**: allow the traffic • **DENY**: deny the traffic.

Parameter	Description
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	Firewall rule name. Example: **name("Rule example")**.
desc	A description of the rule. Example: **desc("Firewall rule example configured in CLI")**.
ips_profile	IDPS profile. For more details about how to create and configure IDPS profiles using CLI, see Configuring IDPS Profiles. Example: **ips_profile("Test ips profile")**.
l7_profile	The applications profile. For more details about how to create and configure applications profiles using CLI, see Configuring Application Profiles. Example: **l7_profile("Test application-profile")**.
reject_with	This setting is available for rules with the **DENY** action: • **reject_with(no)** • **reject_with("host_unreach")**: block traffic and send an "ICMP host unreachable" message • **reject_with("tcp_rst")**: block traffic and send a "TCP connection reset" message **Important!** If **Send TCP reset** is selected, you need to specify a service that uses the TCP protocol (for more details about how to add and configure services, see the section "Configuring Services"). • **reject_with("tcp_reset-both")**: block traffic and send a "TCP connection reset" message to both the client and the server.
scenario	Scenario that needs to be active for the rule to trigger. To specify a scenario: **scenario = "Example of a scenario"**. For more details on configuring scenarios, see Configuring scenarios.
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled.

Parameter	Description
	• **rule_log(yes)** or **rule_log(true)**: log all network packets without setting any limits. To set a limit, you need to specify the number of events to be logged per time unit (s for second, min for minute, h for hour, and d for day; the minimum log limit is 5 packets per day) and the maximum number of packets logged per event. For example, **rule_log(yes, "3/h", 5)** enables logging with the following limits: 3 events per hour with a maximum number of packets per event of 5. • **rule_log(session)**: log the session start.
fragmented	Specify packets to which the firewall rule applies: • **fragmented(yes)** or **fragmented(true)**: apply the rule to fragmented packets only • **fragmented(no)** or **fragmented(false)**: apply the rule to unfragmented packets only • **fragmented(all)**: apply the rule to all packets. If **fragmented** is not specified, the firewall rule is applied to all packets.
src.zone	Traffic source zone. To specify a source zone, such as Trusted: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add source IP address or domain lists. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
user	Users and user groups for which the firewall rule applies (local or LDAP).

Parameter	Description
	To add LDAP groups and users, you need to have a correctly configured LDAP connector (for more information about configuring LDAP connectors via the CLI, see the Configuring LDAP Connectors section). Examples of adding users to a rule: `user = known` `user = "user"` `user = "testd.local\\user1"` `user = ("user", "testd.local\\user1")`
dst.zone	Traffic destination zone. To specify a source zone, such as Untrusted: **dst.zone = Untrusted**. For more details about how to configure zones using CLI, see the Zones section.
dst.ip	Add lists of destination IP addresses or domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
dst.geoip	To specify a destination GeoIP, it is necessary to specify a country code (for example, **dst.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
service	Service type. You can specify a service or a services group (for more details, see Configuring services and Configuring services groups). To specify a single service: **service = "service name"**. To specify multiple services: **service = (service-name1, service-name2, ...)**. To specify a service group: **service = lib.service()**. Provide the services group name in parentheses.

Parameter	Description
time	Set a schedule for a rule. To set a schedule: **time = lib.time()**. Specify a time set group name in parentheses. For more details on configuring time sets, see Configuring time sets.

Example command to create a firewall rule using UPL:

```
Admin@nodename# create network-policy firewall 1 upl-rule PASS \
...src.zone = Trusted \
...dst.zone = Untrusted \
...user = known \
...service = HTTP \
...rule_log(session) \
...name("Test firewall rule") \
...enabled(true)
...
Admin@nodename# show network-policy firewall 1
% ---------------- 1 -----------------
PASS \
    user = known \
    src.zone = Trusted \
    dst.zone = Untrusted \
    service = HTTP \
    rule_log(session) \
    enabled(true) \
    id("1505d309-621b-4f88-a2e4-98667c477535") \
    name("Test firewall rule")
```

Configuring NAT and Routing Rules

You configure NAT and routing rules at the **network-policy nat-routing** level. For more details on the command structure, see UserGate Policy Language.

```
Admin@nodename# create network-policy nat-routing 1 upl-rule
<parameters>
```

Configuring NAT rules

To configure a NAT rule, specify the following parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	NAT rule name. Example: **name("NAT rule example")**.
desc	A description of the rule. Example: **desc("NAT rule example set via CLI")**.
nat	Rule type (specified in the rule properties).
snat_target_ip	IP address with which the source address will be replaced. Specify the address in "", e.g. **snat_target_ip ("1.1.1.1")**.
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled. • **rule_log(session)**: log the session start.
src.zone	Traffic source zone. To specify a source zone, such as Trusted: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add lists of source IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.

Parameter	Description
	To specify source MAC addresses, such as 02:00:00:00:00:00, use **src.ip= 02:00:00:00:00:00**.
dst.zone	Traffic destination zone. To specify a traffic destination zone, such as Untrusted: **dst.zone = Untrusted**. For more details about how to configure zones using CLI, see the Zones section.
dst.ip	Add lists of destination IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify destination MAC addresses, such as 02:00:00:00:00:00, use **dst.ip= 02:00:00:00:00:00**.
service	Service type. You can specify a service or a services group (for more details, see Configuring services and Configuring services groups). To specify a single service: **service = "service name"**. To specify multiple services: **service = (service-name1, service-name2, ...)**. To specify a service group: **service = lib.service()**. Provide the services group name in parentheses.

Example command to create a NAT rule using UPL:

```
Admin@nodename# create network-policy nat-routing 1 upl-rule PASS \
...src.zone = Trusted \
...dst.zone = Untrusted \
...nat \
...rule_log(session) \
...name("Test NAT rule") \
...enabled(true)
...
Admin@nodename# show network-policy nat-routing 1
```

```
% ---------------- 1 ----------------
OK \
    src.zone = Trusted \
    dst.zone = Untrusted \
    direction(input) \
    rule_log(session) \
    enabled(true) \
    id("0344640b-b392-4920-9853-77d85ec1338c") \
    name("Test NAT rule")\
    nat
```

Configuring DNAT Rules

To configure a **DNAT** rule, specify the following parameters.

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	DNAT rule name. Example: **name("DNAT rule example")**.
desc	A description of the rule. Example: **desc("DNAT rule example created via CLI")**.
dnat	Rule type (specified in the rule properties).
snat_target_ip	IP address with which the source address will be replaced. Specify the address in "", e.g. **snat_target_ip ("1.1.1.1")**.
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled. • **rule_log(session)**: log the session start.

Parameter	Description
src.zone	Traffic source zone. To specify a source traffic zone, such as Trusted: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add lists of source IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify source MAC addresses, such as 02:00:00:00:00:00, use **src.ip= 02:00:00:00:00:00**.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
dst.zone	Traffic destination zone. To specify a destination zone, such as Untrusted: **dst.zone = Untrusted**. For more details about how to configure zones using CLI, see the Zones section.
dst.ip	Add lists of destination IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify destination MAC addresses, such as 02:00:00:00:00:00, use **dst.ip= 02:00:00:00:00:00**.
service	

Parameter	Description
	Service type. You can specify a service or a services group (for more details, see Configuring services and Configuring services groups). To specify a single service: **service = "service name"**. To specify multiple services: **service = (service-name1, service-name2, ...)**. To specify a service group: **service = lib.service()**. Provide the services group name in parentheses.
target_ip	DNAT destination address. To specify a destination address: **target_ip("1.1.1.1")**.
target_snat	Replace the source IP address with the UserGate address: • **target_snat(yes)** or **target_snat(true)** • **target_snat(no)** or **target_snat(false)**

Example command to create a DNAT rule using UPL:

```
Admin@nodename# create network-policy nat-routing 1 upl-rule PASS \
...src.zone = Untrusted \
...target_ip("10.10.0.15") \
...dnat \
...rule_log(session) \
...name("Test DNAT") \
...enabled(yes)
...
Admin@nodename# show network-policy nat-routing 1
% ---------------- 1 -----------------
OK \
    src.zone = Untrusted \
    target_ip("10.10.0.15") \
    direction(input) \
    rule_log(session) \
    enabled(true) \
    id("00e60d4e-9b93-454b-a424-58e2102f84c2") \
    name("Test DNAT")\
    dnat
```

Configuring port forwarding rules

To configure a **Port forwarding** rule, specify the following parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	Port forwarding rule name. Example: **name("Port forwarding rule example")**.
desc	A description of the rule. Example: **desc("Port forwarding rule example created via CLI")**.
port_mapping	Rule type (specified in the rule properties).
snat_target_ip	IP address with which the source address will be replaced. Specify the address in "", e.g. **snat_target_ip ("1.1.1.1")**.
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled. • **rule_log(session)**: log the session start.
src.zone	Traffic source zone. Example source zone: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add lists of source IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.

Parameter	Description
	To specify source MAC addresses, such as 02:00:00:00:00:00, use **src.ip= 02:00:00:00:00:00**.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
dst.ip	Add lists of destination IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify destination MAC addresses, such as 02:00:00:00:00:00, use **dst.ip= 02:00:00:00:00:00**.
port_map	Port overrides for published services. To override, specify the network protocol (TCP, UDP, SMTP, SMTPS), and the original and the new destination ports. Example: **port_map(tcp, 2000, 2100)**. **Important!** The ports listed here may not be used as they are reserved for UserGate's internal services: 2200, 8001, 4369, 9000-9100.
target_ip	DNAT destination address. To specify a destination address: **target_ip("1.1.1.1")**.
target_snat	Replace the source IP address with the UserGate address: • **target_snat(yes)** or **target_snat(true)** • **target_snat(no)** or **target_snat(false)**

Example command to create a port forwarding rule using UPL:

```
Admin@nodename# create network-policy nat-routing 8 upl-rule OK \
... src.zone = Untrusted \
```

```
... dst.ip = lib.network(UG_IP) \
... target_ip("10.10.0.16") \
... port_map(tcp, 2222, 23) \
... rule_log(session) \
... name(port_fw1) \
... port_mapping \
...
Admin@nodename# show network-policy nat-routing 8
% ---------------- 8 ----------------
OK \
    src.zone = Untrusted \
    dst.ip = lib.network(UG_IP) \
    target_ip("10.10.0.16") \
    port_map(tcp, 2222, 23) \
    direction(input) \
    rule_log(session) \
    enabled(true) \
    id("1af47c3f-96a3-4e65-90e3-debf169bb745") \
    name(port_fw1)\
    port_mapping
```

Configuring Policy-based routing rules

To configure a **Policy-based routing** rule, specify the following parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	Policy-based routing rule name. Example: **name("Policy-based routing rule example")**.
desc	A description of the rule. Example: **desc("Policy-based routing rule example set via CLI")**.

Parameter	Description
route	Rule type (specified in the rule properties).
gateway	Select one of the existing gateways: **gateway("1.1.1.1")**. For more details about adding a gateway using CLI, see Gateway Configuration.
scenario	Scenario that needs to be active for the rule to trigger. To specify a scenario: **scenario = "Example of a scenario"**. For more details on configuring scenarios, see Configuring scenarios.
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled. • **rule_log(session)**: log the session start.
src.zone	Traffic source zone. Example source zone: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add lists of source IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify source MAC addresses, such as 02:00:00:00:00:00, use **src.ip= 02:00:00:00:00:00**.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
dst.ip	Add lists of destination IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to

Parameter	Description
	create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify destination MAC addresses, such as 02:00:00:00:00:00, use **dst.ip= 02:00:00:00:00:00**.
dst.geoip	To specify a destination GeoIP, it is necessary to specify a country code (for example, **dst.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
service	Service type. You can specify a service or a services group (for more details, see Configuring services and Configuring services groups). To specify a single service: **service = "service name"**. To specify multiple services: **service = (service-name1, service-name2, ...)**. To specify a service group: **service = lib.service()**. Provide the services group name in parentheses.
user	Users and user groups for which the rule applies (local or LDAP). To add LDAP groups and users, you need to have a correctly configured LDAP connector (for more information about configuring LDAP connectors via the CLI, see Configuring LDAP connectors). Examples of adding users to a rule: `user = known` `user = "user"` `user = "testd.local\\user1"` `user = ("user", "testd.local\\user1")`

Example of creating and adding a policy-based routing rule using UPL:

```
Admin@nodename# create network-policy nat-routing 7 upl-rule OK \
... route \
```

```
... gateway("def") \
... name("testpbr1") \
... enabled(true) \
... rule_log(session) \
...
Admin@nodename# set network-policy nat-routing 7 upl-rule OK \
... service = (HTTPS, HTTP) \
...
Admin@nodename# set network-policy nat-routing 7 upl-rule OK \
... user = "CN=Users1,DC=LOCAL"
Admin@nodename# show network-policy nat-routing 7
% ---------------- 7 -----------------
OK \
    user = "CN=Users1,DC=LOCAL" \
    service = (HTTPS, HTTP) \
    gateway(def) \
    direction(input) \
    rule_log(session) \
    enabled(true) \
    id("0585a95f-4707-4c11-840d-44643bc2c799") \
    name(testpbr1)\
    route
```

Configuring Network mapping rules

To configure a Network mapping rule, specify the following parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	Network mapping rule name. Example: **name("Network mapping rule example")**.
desc	A description of the rule.

Parameter	Description
	Example: **desc("Network mapping rule example set via CLI")**.
netmap	Rule type (specified in the rule properties).
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled. • **rule_log(session)**: log the session start.
src.zone	Traffic source zone. Example source zone: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add lists of source IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. To specify source MAC addresses, such as 02:00:00:00:00:00, use **src.ip= 02:00:00:00:00:00**.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
dst.ip	Add lists of destination IP addresses, MAC addresses, and domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.

Parameter	Description
	To specify destination MAC addresses, such as 02:00:00:00:00:00, use **dst.ip= 02:00:00:00:00:00**.
dst.geoip	To specify a destination GeoIP, it is necessary to specify a country code (for example, **dst.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
service	Service type. You can specify a service or a services group (for more details, see Configuring services and Configuring services groups). To specify a single service: **service = "service name"**. To specify multiple services: **service = (service-name1, service-name2, ...)**. To specify a service group: **service = lib.service()**. Provide the services group name in parentheses.
target_ip	Parameter for network substitution: address of a network to use in the substitution. Example: **target_ip("1.1.1.0")**.
direction	Parameter for network substitution. Direction: • **direction(input)**: input, replace the destination IP network address. destination IP addresses in the traffic that matches the rule conditions will be substituted. The network address is replaced with the network specified in the value **target_ip**. • **direction(output)**: output, replace the source IP network address. source IP addresses in the traffic that matches the rule conditions will be substituted. The network address is replaced with the network specified in the value **target_ip**.

Example command to create a network mapping rule using UPL:

```
Admin@nodename# create network-policy nat-routing 8 upl-rule OK \
... src.zone = External \
... target_ip("192.168.222.0/24") \
... direction(output) \
... netmap \
... rule_log(session) \
... name(netmap1) \
...
```

```
Admin@nodename# show network-policy nat-routing 8
% ---------------- 8 ----------------
OK \
    src.zone = External \
    target_ip("192.168.222.0/24") \
    direction(output) \
    rule_log(session) \
    enabled(true) \
    id("26cbd3e8-0210-494c-9fd4-57300b47a9fe") \
    name(netmap1)\
    netmap
```

Configuring Load Balancing

Load balancing rules are configured at the **network-policy load-balancing** level using UPL policies. For more details on the command structure, see UserGate Policy Language.

Load balancing settings for TCP / UDP will be discussed below.

To display information about all load balancers, use the following command:

```
Admin@nodename# show network-policy load-balancing
```

Configuring TCP/UDP load balancers

You configure this section at the **network-policy load-balancing tcp-udp** level.

To create a TCP/UDP load balancer, use the following command:

```
Admin@nodename# create network-policy load-balancing tcp-udp <position>
upl-rule
```

TCP/UDP load balancing rules have the following parameters:

Parameter	Description
PASS	Action to create a rule using UPL.

Parameter	Description
OK	
name	The name of the balancing rule. Example: **name("TCP_UDP balancer")**.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
desc	A description of the rule. Example: **desc("TCP_UDP balancing- rule")**.
src.zone	Traffic source zone. To specify a source zone, such as Trusted: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add source IP address or domain lists. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section. Example: **src.ip = lib.network("Test ip-list")**.
src.geoip	Specify a Geo IP as the source. Example: **src.geoip = US**.
url.address	Virtual server IP address. Example: **url.address = 10.10.0.20**.
url.port	The port for which load balancing is to be performed. Example: **url.port = 1812**.
service	The protocol (TCP or UDP) for which load balancing is to be performed. Example: **service = udp**.
scheduler	

Parameter	Description
	Load balancing methods for real servers: • **rr (round robin)**: each new connection is passed to the next server in the list, loading all servers evenly. • **wrr (weighted round robin)**: similar to round robin, but the real servers are loaded taking their weights into account, which allows you to distribute the load allowing the performance of each server to be taken into account. • **lc (least connections)**: a new connection is sent to the server which currently has the least number of connections. • **wlc (weighted least connections)**: similar to least connections, but the real servers are loaded taking their weights into account, which allows the performance of each server to be taken into account. Example: **scheduler(rr)**.
real_server	Real servers to which traffic will be redirected. You need to specify the following for a server: • **ip**: the server's IP address • **port**: the server port to which requests from users will be redirected • **weight**: the weight to be used for uneven load distribution on real servers • **mode**: the operating mode: ◦ **gate** (gateway mode): use routing to redirect traffic to the virtual server ◦ **masq** (masquerading mode): DNAT is used to forward the traffic to the virtual server ◦ **masq-snat** (masquerading mode with the source IP overridden): similar to the masq mode, but UserGate will substitute the source IP address with its own. Example: **real_server(masq, 10.10.0.9:1812, 50)**.
ipvs_fallback	Configure fallback: • **ip**: the server's IP address • **port**: the server port to which requests from users will be forwarded • **mode**: the operating mode: ◦ **gate** (gateway mode): use routing to redirect traffic to the virtual server

Parameter	Description
	◦ **masq** (masquerading mode): DNAT is used to forward the traffic to the virtual server ◦ **masq-snat** (masquerading mode with the source IP overridden): similar to the masq mode, but UserGate will substitute the source IP address with its own. Example: **ipvs_fallback(masq, 10.10.100.100:1812)**.
monitor	Configure real server monitoring: • **kind**: the checking type ◦ **ping**: check if the node is reachable using the ping utility. ◦ **connect**: check if the node is up and running by establishing a TCP connection to a specific port. ◦ **negotiate**: check node health by sending a certain HTTP or DNS request and comparing the response against the expected one. • **service**: specify the service (**HTTP** or **DNS**) if the checking type is **negotiate**. • **request**: must be specified if the checking type is **negotiate**. • **response**: the expected response. Must be specified if the checking type is **negotiate**. • **interval**: the time interval at which checks should be performed. • **timeout**: how long to wait for the response. • **max-failures**: the maximum number of attempts to check real servers, after which a server is considered inoperable and excluded from balancing. Example: `monitor_kind(ping) \` `monitor_interval(60) \` `monitor_timeout(60) \` `monitor_failurecount(10) \`

To edit an existing load balancing rule, use the following command:

```
 Admin@nodename# set network-policy load-balancing tcp-udp <position>
upl-rule
```

To display information about all TCP/UDP balancing rules, use the following command:

```
 Admin@nodename# show network-policy load-balancing tcp-udp
```

To display information about a specific TCP/UDP load balancing rule, use the following command:

```
Admin@nodename# show network-policy load-balancing tcp-udp <position>
```

Example command to create a load balancing rule using UPL:

```
Admin@nodename# create network-policy load-balancing tcp-udp 1 upl-rule
OK \
...src.zone = Trusted \
...url.address = 10.10.0.20 \
...url.port = 1812 \
...service = udp \
...scheduler(rr) \
...real_server((gate, 10.10.0.9, 50), (gate, 10.10.0.8, 50)) \
...name(tcpudp_balancer1) \
...enabled(true)
...
Admin@nodename# show network-policy load-balancing tcp-udp

% ----------------- 1 -----------------
OK \
    src.zone = Trusted \
    url.address = 10.10.0.20 \
    url.port = 1812 \
    service = udp \
    scheduler(rr) \
    real_server((gate, 10.10.0.9, 50), (gate, 10.10.0.8, 50)) \
    monitor_kind(ping) \
```

```
    monitor_interval(60) \
    monitor_timeout(60) \
    monitor_failurecount(10) \
    enabled(true) \
    id(cbed6ed7-901e-4641-83a1-a05f82dae177) \
    name(tcpudp_balancer1)
```

To delete an existing load balancer, use the following command:

```
Admin@nodename# delete network-policy load-balancing tcp-udp <position>
```

Configuring Traffic Shaping Rules

Traffic shaping rules are configured at the **network-policy traffic-shaping** level using the UPL language syntax. For more details on the command structure, see UserGate Policy Language.

To create a traffic shaping rule, use the following command:

```
Admin@nodename# create network-policy traffic-shaping <position> upl-
rule
```

Traffic shaping rule settings:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	Traffic shaping rule name. Example: **name("Traffic shaping rule example")**.
desc	A description of the rule.

Parameter	Description
	Example: **desc("The example of traffic shaping rule configured in CLI")**.
bandwidth_pool	The bandwidth pool, e.g., **bandwidth_pool("1 Mbps")**. For more details about creating and configuring bandwidth pools, see Configuring Bandwidth Pools.
scenario	Scenario that needs to be active for the rule to trigger. To specify a scenario: **scenario = "Example of a scenario"**. For more details on configuring scenarios, see Configuring scenarios.
rule_log	Log traffic information if the rule is triggered. The available options are: • **rule_log(no)** or **rule_log(false)**: disable logging. If **rule_log** is not specified, logging is disabled. • **rule_log(yes)** or **rule_log(true)**: log all network packets without setting any limits. To set a limit, you need to specify the number of events to be logged per time unit (s for second, min for minute, h for hour, and d for day; the minimum log limit is 5 packets per day) and the maximum number of packets logged per event. For example, **rule_log(yes, "3/h", 5)** enables logging with the following limits: 3 events per hour with a maximum number of packets per event of 5. • **rule_log(session)**: log the session start.
src.zone	Traffic source zone. To specify a source zone, such as Trusted: **src.zone = Trusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Add source IP address or domain lists. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
src.geoip	Source GeoIP. Specify a country code (for example, **src.geoip = AE**). Click here for the list of ISO 3166-1 country codes.

Parameter	Description
	Important! The maximum number of GeoIPs that can be specified is limited to 15.
user	Users and user groups for which the traffic shaping rule applies (local or LDAP). To add LDAP groups and users, you need to have a correctly configured LDAP connector (for more information about configuring LDAP connectors via the CLI, see the Configuring LDAP Connectors section). Examples of adding users to a traffic shaping rule: `user = known` `user = "user"` `user = "testd.local\\user1"` `user = ("user", "testd.local\\user1")`
dst.zone	Traffic destination zone. To specify the destination zone, use: **dst.zone = Untrusted**. For more details about how to configure zones using CLI, see the Zones section.
dst.ip	Add lists of destination IP addresses or domains. To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a destination domain list: **dst.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
dst.geoip	To specify a destination GeoIP, it is necessary to specify a country code (for example, **dst.geoip = AE**). Click here for the list of ISO 3166-1 country codes. **Important!** The maximum number of GeoIPs that can be specified is limited to 15.
service	Service type. You can specify a service or a services group (for more details, see Configuring services and Configuring services groups). To specify a single service: **service = "service name"**. To specify multiple services: **service = (service-name1, service-name2, ...)**.

Parameter	Description
	To specify a service group: **service = lib.service()**. Provide the services group name in parentheses.
application	List of applications to which this rule applies. You can specify: • All application groups: **application = lib.category(All)**. • Specific application group: **application = lib.applicationgroup()**. Provide the application group name in parentheses. • Application categories: **application = lib.category()**. Provide the application category name in parentheses.
time	Set a schedule for a rule. To set a schedule: **time = lib.time()**. Specify a time set group name in parentheses. For more details on configuring time sets, see Configuring time sets.

To edit a traffic shaping rule, use the following command:

```
Admin@nodename# set network-policy traffic-shaping <position> upl-rule
```

To view all traffic shaping rules, use the following command:

```
Admin@nodename# show network-policy traffic-shaping
```

To view a specific traffic shaping rule, use the following command:

```
Admin@nodename# show network-policy traffic-shaping <position>
```

Example command to create a traffic shaping rule using UPL:

```
Admin@nodename# create network-policy traffic-shaping 1 upl-rule OK \
...user = known \
...src.zone = Trusted \
...dst.zone = Untrusted \
...service = (HTTP, HTTPS) \
...time = lib.time("Working hours") \
...rule_log(session) \
```

```
...bandwidth_pool("1 Mbps") \
...name("Test traffic shaping rule") \
...desc("Test traffic shaping rule description") \
...enabled(true)
...
Admin@nodename# show network-policy traffic-shaping 1

% ----------------- 1 -----------------
OK \
    user = known \
    src.zone = Trusted \
    dst.zone = Untrusted \
    service = (HTTP, HTTPS) \
    time = lib.time("Working hours") \
    desc("Test traffic shaping rule description") \
    rule_log(session) \
    bandwidth_pool("1 Mbps") \
    enabled(true) \
    id(e63c34e6-af7f-4a4d-a29d-b51d4070655c) \
    name("Test traffic shaping rule")
```

To delete a traffic shaping rule, use the following command:

```
Admin@nodename# delete network-policy traffic-shaping <position>
```

CONFIGURING REMOTE ACCESS (VPN)

Configuring Server Rules

You configure server rules at the **vpn server-rules** level. For more details on the structure of commands used to configure server rules, please read the UserGate Policy Language section.

To create an VPN server rule, use the following command:

```
Admin@nodename# create vpn server-rules <position> upl-rule
<parameters>
```

You need to specify the following parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**. If not specified when it is created, the rule will be enabled once created.
name	VPN server rule name. Example: **name("VPN server rule example")**.
desc	A description of the rule. Example: **desc("VPN server rule example configured in CLI")**.
profile	VPN security profile that defines a pre-shared encryption key and algorithms for encryption and authentication. Example: **profile("Client VPN profile")**. For more details on configuring security profiles, see the Configuring VPN Security Profiles section.
vpn_network	VPN network. To specify a network: **vpn_network("VPN network example")**. For more details about how to configure VPN using CLI, see the Configuring VPN Network section.
auth_profile	Authentication profile for VPN users. The same authentication profile may be used that you use to authorize users for Internet access. Note that transparent authentication methods such as Kerberos, NTLM, or SAML IDP cannot be used for VPN authorization. To specify a authentication profile: **auth_profile("Example user auth profile")**. For more details about how to create and configure auth profiles using CLI, see the Configuring Authentication Profiles section.

Parameter	Description
interface	VPN interface to connect VPN clients. To specify an interface, for example, tunnel1: **interface(tunnel1)**. For more information about how to add and configure VPN interfaces, see the VPN Device Settings section.
ep_only	This option (available starting with software release 7.1.2) allows you to limit the connection possibility according to this rule only for UserGate Client VPN clients (**true**, **false**).
src.zone	Zone from which VPN connections are allowed. To specify a source zone, such as Untrusted: **src.zone = Untrusted**. For more details about how to configure zones using CLI, see the Zones section.
src.ip	Lists of IP addresses or domains from which VPN connections are allowed. To specify a list of IP addresses: **src.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section. To specify a source domain list: **src.ip = lib.url()**. Provide the URL to which the desired domains were added in parentheses. For more details about how to create and configure URL lists using the CLI, see the Configuring URL Lists section.
user	Users and user groups allowed to connect via VPN. To add LDAP groups and users, you need to have a correctly configured LDAP connector (for more information about configuring LDAP connectors via the CLI, see the Configuring LDAP Connectors section). The following line describes how to add a local user (local_user) and group (Local Group), a user (example.local\AD_user), and an LDAP group (AD group): `user = (local_user, "CN=Local Group,DC=LOCAL", "example.loc\\AD_user", "CN=AD group,OU=Example,DC=example,DC=loc")` The Active Directory domain example.loc has been already configured. When adding LDAP users and groups, you can specify a list of paths on the server, starting from which the system will search for users and groups.
dst.ip	Lists of IP addresses of the interface to which the clients will be connected.

Parameter	Description
	To specify a list of IP addresses: **dst.ip = lib.network()**. Provide the list name in parentheses. For more details about how to create and configure IP address lists using CLI, see the Configuring IP addresses section.

Example of creating a VPN server rule:

```
Admin@nodename# create vpn server-rules 3 upl-rule OK\
...name("Test server VPN rule") \
...desc("Test server VPN rule description") \
...profile("New server VPN profile") \
...vpn_network("Test VPN network") \
...auth_profile(Local) \
...interface(tunnel3) \
...src.zone = Untrusted \
...dst.ip = lib.network("UG address") \
...user = ("CN=VPN servers,DC=LOCAL") \
...enabled(true) \
```

To edit a VPN server rule, use the following command:

```
Admin@nodename# set vpn server-rules <position> upl-rule <parameters>
```

To remove a VPN server rule, use the following command:

```
Admin@nodename# delete vpn server-rules <position>
```

To view the configured VPN server rules, use the following command:

```
Admin@nodename# show vpn server-rules <position>
```

Configuring client rules

You configure client rules at the **vpn client-rules** level. For more details on the structure of commands used to configure client rules, please read the UserGate Policy Language.

To create a VPN client rule, use the following command:

```
Admin@nodename# create vpn client-rules <position> upl-rule
<parameters>
```

You need to specify the following parameters:

Parameter	Description
PASS **OK**	Action to create a rule using UPL.
enabled	Enable/disable a rule: • **enabled(yes)** or **enabled(true)**. • **enabled(no)** or **enabled(false)**.
name	VPN client rule name. Example: **name("VPN client rule example")**.
desc	VPN client rule description. Example: **desc("VPN client rule example set in CLI")**.
profile	VPN security profile that defines a pre-shared encryption key and algorithms for encryption and authentication. Example: **profile("Client VPN profile")**. For more details on configuring security profiles, see the Configuring VPN Security Profiles section.
interface	VPN interface to connect VPN clients. To specify an interface, for example, tunnel1: **interface(tunnel1)**. For more information about how to add and configure VPN interfaces, see the VPN Device Settings section.
server_address	IP address of the VPN server to which this VPN client connects. It is usually the IP address of an interface in the Untrusted zone on NGFW that acts as a VPN server. Format: **server_address("1.2.3.4")**.

When displaying rules, the last VPN error, the connection status, and the connection time will be displayed in addition to the specified conditions and properties.

Example of creating a VPN client rule:

```
Admin@nodename# create vpn client-rules 2 upl-rule OK\
...name("Test VPN client rule") \
...desc("Test VPN client rule description") \
...profile("Client VPN profile") \
...interface(tunnel3) \
...server_address("10.10.0.1") \
...enabled(true) \
```

To edit a VPN client rule, use the following command:

```
Admin@nodename# set vpn client-rules <position> upl-rule <parameters>
```

To remove a VPN client rule, use the following command:

```
Admin@nodename# delete vpn client-rules <position>
```

To view parameters for VPN client rules that were created, use the following command:

```
Admin@nodename# show vpn client-rules <position>
```

Configuring a VPN Network

You configure VPN networks at the **vpn networks** level.

To create a VPN network, use the following command:

```
Admin@nodename# create vpn networks <parameters>
```

VPN network parameters:

Parameter	Description
name	VPN network name.
description	VPN network description.
ip-range	Range of IP addresses to be used by the clients and the server. Format: <IP_start-IP_end> Exclude the addresses assigned to the VPN interface used with this network from the range. Do not enter network addresses or the broadcast address here.
mask	Subnet mask, e.g. 255.255.255.0.
use-system-dns	Assign DNS servers used by UserGate to the client: • **on**: use system DNS servers • **off**: do not use system DNS servers
dns-servers	DNS servers that will be passed to the client.
routes-ip	VPN route. Specify an IP address in the following formats: "A.B.C.D" or "A.B.C.D/m".
routes-ip-list	VPN route. Specify a group of IP addresses. For more details on creating IP address groups using CLI, see the Configuring IP Addresses section.
all-routes	No VPN connection routing restrictions when using the UserGate VPN client.
include-routes-ip	IP addresses access to which should be routed via VPN connection when using the UserGate VPN client.
include-routes-ip-list	IP address list access to which should be routed via VPN connection when using the UserGate VPN client.
exclude-routes-ip	IP addresses access to which should be disabled via VPN connection when using the UserGate VPN client.
exclude-routes-ip-list	IP address list access to which should be disabled via VPN connection when using the UserGate VPN client.
restrict-lan-access	Restrict access to the local network when using the UserGate VPN client.

Example of creating a VPN network:

```
Admin@nodename# create vpn networks name "Test VPN network" description
"This is a new test VPN network" ip-range 10.10.3.2-10.10.2.200 mask
255.255.255.0
```

To edit network parameters, use the following command:

```
Admin@nodename# set vpn networks <network-name> <parameters>
```

To delete a VPN network or individual parameters of a network, use the following command:

```
Admin@nodename# delete vpn networks <network-name>
```

To display information about a VPN network, use the following command:

```
Admin@nodename# show vpn networks <network-name>
```

Configuring VPN security profiles

Starting from nodename 7.1.0 two types of VPN security profiles are defined: server profiles and client profiles.

VPN security profiles are configured at the **vpn server-security-profiles** and the **vpn client-security-profiles** levels.

Creating a VPN Server Security Profile

To create a VPN server security profile, use the following command:

```
Admin@nodename# create vpn server-security-profiles <parameters>
```

VPN server security profile parameters:

Parameter	Description
name	VPN security profile name.

Parameter	Description
description	VPN security profile description.
protocol	The version of a protocol used to create a secure link between two networks. The options are as follows: • **ipsec**: IPsec(IKEv1). • **ipsec-l2tp**: IPsec(IKEv1)/L2TP. • **ikev2**: IPsec(IKEv2).
ike-mode	IKE mode: • **main**: the main mode. In the main mode, the devices exchange six messages. During the first exchange (messages 1 and 2), the encryption and authentication algorithms are negotiated. The second exchange (messages 3 and 4) implements the Diffie-Hellman (DH) key exchange. After the second exchange, the IKE service on each device creates a master key to use for authentication. The third exchange (messages 5 and 6) authenticates the reporter and responder of the connection (identity checking) and the information is secured using the encryption algorithm established earlier. • **aggressive**: the aggressive mode. In the aggressive mode, there are 2 exchanges, 3 messages in total. In the first message, the reporter transmits information corresponding to messages 1 and 3 of the main mode — that is, the information on encryption and authentication algorithms as well as the DH key. The second message, transmitted by the responder, contains information corresponding to messages 2 and 4 of the main mode and also authenticates the responder. The third message authenticates the reporter and confirms the exchange.
local-id-type	IKE local ID parameter type. Required for peer node validation when establishing a VPN connection using hardware from some vendors. Enumerated parameter options: • **none**: field default value. Used when the IKE local ID parameter is not required for establishing a VPN connection. For example, when a VPN connection between two UserGate nodes is established. • **IPv4**: the host's IP address. • **FQDN**: the host's address in the fully-qualified domain name (FQDN) format. • **CIDR**: the host's address in the classless inter-domain routing (CIDR) format.

Parameter	Description
local-id-value	IKE local ID parameter value in selected format.
psk	Pre-shared key. Used to authenticate a remote node using pre-shared key. This string must match on the client and server for a successful connection.
certificate	VPN server certificate for authentication via certificate.
authentication-mode	Authentication method. It is possible to authenticate using a login and password via the EAP (AAA) method, or via certificates (PKI).
user-certificate-profile	When choosing PKI authentication method it is necessary to specify a previously configured client certificate profile.
phase1-key-lifetime	Key lifetime: the time period after which the parties re-authenticate and re-negotiate the first-phase settings.
dpd-state	Operating modes of the Dead Peer Detection mechanism, which checks the functionality of the VPN channel and promptly disconnects/reconnects it when the connection is lost. There are 3 possible operating modes of the mechanism: • **off**: the mechanism is disabled. DPD requests are not sent. • **always**: DPD requests are always sent within the specified time interval. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **dpd-max-failures** parameter. If there is a response, the mechanism returns to the initial interval for sending DPD requests, and if there is no response, the connection is terminated. • **idle**: DPD requests are not sent while there is ESP traffic through the created SAs. If there are no packets within twice the specified time interval, then a DPD request is sent. If there is a response, a new DPD request will be sent again after a double interval of the specified time. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **dpd-max-failures** parameter. If there is no response, the connection is terminated.

Parameter	Description
dpd-interval	Dead Peer Detection mechanism checking interval. Minimum interval: 10 seconds. The Dead Peer Detection (DPD) mechanism is used to perform a health check and availability check of neighbor devices. DPD periodically sends R-U-THERE messages to check the availability of the IPsec neighbor (default value: 60 seconds).
dpd-max-failures	Maximum number of unreachable IPsec neighbor detection requests to be sent before an IPsec neighbor is considered unreachable (default value: 5).
dh-groups	Diffie-Hellman groups to be used for key exchange. Instead of the key itself, certain general information is transmitted that the DH key generation algorithm needs to create the shared secret key. The larger the Diffie-Hellman group number, the more bits are used to make the key secure. • **Group 1 Prime 768 bit** • **Group 2 Prime 1024 bit** • **Group 5 Prime 1536 bit** • **Group 14 Prime 2048 bit** • **Group 15 Prime 3072 bit** • **Group 16 Prime 4096 bit** • **Group 17 Prime 6144 bit** • **Group 18 Prime 8192 bit**
phase1-security	Authentication and encryption algorithms. To specify authentication and encryption algorithms, use the following command: `Admin@nodename# create vpn server-security-profiles ... phase1-security new auth-alg <auth-alg-name> encrypt-alg <encrypt-alg-name>` Available values: • **auth-alg**: select an authentication algorithm. ◦ **MD5** ◦ **SHA1** ◦ **SHA256** ◦ **SHA384** ◦ **SHA512**

Parameter	Description
	• **encrypt-alg**: select an encryption algorithm. ◦ **DES** ◦ **3DES** ◦ **AES128** ◦ **AES192** ◦ **AES256**
phase2-key-lifetime	Key lifetime: the time period after which the nodes must rotate the encryption key. The lifetime for the second phase is shorter than for the first one, which entails a more frequent key rotation.
key-lifesize-enabled	Enable configuration mode with the maximum data size encrypted by one key.
key-lifesize	Maximum key lifesize (in kilobytes). If both values (**phase2-key-lifetime** and **key-lifesize**) are set, the counter that first reaches the limit will trigger re-creating the session keys. To disable the restriction, specify: **off**.
nat-keepalive	NAT keepalive packet sending period in seconds (can be set to 0 or to a value greater than 4). Used in scenarios when IPSec traffic goes through a NAT node. NAT table entries are active for a limited time. If there was no VPN traffic over the tunnel during that time span, NAT table entries on the NAT host will be deleted, preventing further passage of VPN traffic. The VPN server located behind the NAT gateway uses NAT keepalive function to periodically send keepalive packets to a peer node in order to keep the NAT session active.
phase2-security	Authentication and encryption algorithms. To specify authentication and encryption algorithms, use the following command: `Admin@nodename# create vpn server-security-profiles ... phase2-security new auth-alg <auth-alg-name> encrypt-alg <encrypt-alg-name>` Available values: • **auth-alg**: select an authentication algorithm. ◦ **MD5** ◦ **SHA1** ◦ **SHA256** ◦ **SHA384**

Parameter	Description
	◦ **SHA512** • **encrypt-alg**: select an encryption algorithm. ◦ **DES** ◦ **3DES** ◦ **AES128** ◦ **AES192** ◦ **AES256**

Creating a VPN Client Security Profile

To create a VPN client security profile, use the following command:

```
Admin@nodename# create vpn client-security-profiles <parameter>
```

VPN client security profile parameters:

Parameter	Description
name	VPN security profile name.
description	VPN security profile description.
protocol	Protocol for establishing a VPN channel. The options are as follows: • **ipsec-l2tp**: IPsec(IKEv1)/L2TP VPN. • **ipsec**: IPsec (IKEv1) VPN with third-party hardware. • **ikev2**: IPsec(IKEv2) VPN.
ike-mode	IKE mode: • **main**: the main mode. In the main mode, the devices exchange six messages. During the first exchange (messages 1 and 2), the encryption and authentication algorithms are negotiated. The second exchange (messages 3 and 4) implements the Diffie-Hellman (DH) key exchange. After the second exchange, the IKE service on each device creates a master key to use for authentication. The third exchange (messages 5 and 6) authenticates the reporter and responder of the connection (identity checking) and the information is secured using the encryption algorithm established earlier.

Parameter	Description
	• **aggressive**: the aggressive mode. In the aggressive mode, there are 2 exchanges, 3 messages in total. In the first message, the reporter transmits information corresponding to messages 1 and 3 of the main mode — that is, the information on encryption and authentication algorithms as well as the DH key. The second message, transmitted by the responder, contains information corresponding to messages 2 and 4 of the main mode and also authenticates the responder. The third message authenticates the reporter and confirms the exchange.
local-id-type	IKE local ID parameter type. Required for peer node validation when establishing a VPN connection using hardware from some vendors. Enumerated parameter options: • **none**: field default value. Used when the IKE local ID parameter is not required for establishing a VPN connection. For example, when a VPN connection between two UserGate nodes is established. • **IPv4**: the host's IP address. • **FQDN**: the host's address in the fully-qualified domain name (FQDN) format. • **CIDR**: the host's address in the classless inter-domain routing (CIDR) format.
local-id-value	IKE local ID parameter value in selected format.
psk	Pre-shared key. Used to authenticate a remote node using pre-shared key. This string must match on the client and server for a successful connection.
authentication-login	Login previously created at the VPN server to authenticate a node acting as VPN client.
authentication-password	Password created at the VPN server to authenticate a node acting as VPN client.
certificate	VPN server certificate for authentication via certificate.
vpn-local-network	IP address of an allowed local subnet which is used to establish a VPN connection with Cisco node.
vpn-remote-network	IP address of an allowed subnet at the side of a remote VPN server which is used to establish a VPN connection with Cisco node.
phase1-key-lifetime	

Parameter	Description
	Key lifetime: the time period after which the parties re-authenticate and re-negotiate the first-phase settings.
dpd-state	Operating modes of the Dead Peer Detection mechanism, which checks the functionality of the VPN channel and promptly disconnects/reconnects it when the connection is lost. There are 3 possible operating modes of the mechanism: • **off**: the mechanism is disabled. DPD requests are not sent. • **always**: DPD requests are always sent within the specified time interval. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **dpd-max-failures** parameter. If there is a response, the mechanism returns to the initial interval for sending DPD requests, and if there is no response, the connection is terminated. • **idle**: DPD requests are not sent while there is ESP traffic through the created SAs. If there are no packets within twice the specified time interval, then a DPD request is sent. If there is a response, a new DPD request will be sent again after a double interval of the specified time. If no response is received, additional requests are sent sequentially at intervals of 5 seconds in the number specified in the **dpd-max-failures** parameter. If there is no response, the connection is terminated.
dpd-interval	Dead Peer Detection mechanism checking interval. Minimum interval: 10 seconds. The Dead Peer Detection (DPD) mechanism is used to perform a health check and availability check of neighbor devices. DPD periodically sends R-U-THERE messages to check the availability of the IPsec neighbor (default value: 60 seconds).
dpd-max-failures	Maximum number of unreachable IPsec neighbor detection requests to be sent before an IPsec neighbor is considered unreachable (default value: 5).
dh-groups	Diffie-Hellman groups to be used for key exchange. Instead of the key itself, certain general information is transmitted that the DH key generation algorithm needs to create the shared secret key. The larger the Diffie-Hellman group number, the more bits are used to make the key secure. • **Group 1 Prime 768 bit** • **Group 2 Prime 1024 bit** • **Group 5 Prime 1536 bit** • **Group 14 Prime 2048 bit**

Parameter	Description
	• **Group 15 Prime 3072 bit** • **Group 16 Prime 4096 bit** • **Group 17 Prime 6144 bit** • **Group 18 Prime 8192 bit**
phase1-security	Authentication and encryption algorithms. To specify authentication and encryption algorithms, use the following command: `Admin@nodename# create vpn client-security-profiles ... phase1-security new auth-alg <auth-alg-name> encrypt-alg <encrypt-alg-name>` Available values: • **auth-alg**: select an authentication algorithm. ◦ **MD5** ◦ **SHA1** ◦ **SHA256** ◦ **SHA384** ◦ **SHA512** • **encrypt-alg**: select an encryption algorithm. ◦ **DES** ◦ **3DES** ◦ **AES128** ◦ **AES192** ◦ **AES256**
phase2-key-lifetime	Key lifetime: the time period after which the nodes must rotate the encryption key. The lifetime for the second phase is shorter than for the first one, which entails a more frequent key rotation.
key-lifesize-enabled	Enable configuration mode with the maximum data size encrypted by one key.
key-lifesize	Maximum key lifesize (in kilobytes). If both values (**phase2-key-lifetime** and **key-lifesize**) are set, the counter that first reaches the limit will trigger re-creating the session keys. To disable the restriction, specify: **off**.
nat-keepalive	NAT keepalive packet sending period in seconds (can be set to 0 or to a value greater than 4). Used in scenarios when IPSec

Parameter	Description
	traffic goes through a NAT node. NAT table entries are active for a limited time. If there was no VPN traffic over the tunnel during that time span, NAT table entries on the NAT host will be deleted, preventing further passage of VPN traffic. The VPN server located behind the NAT gateway uses NAT keepalive function to periodically send keepalive packets to a peer node in order to keep the NAT session active.
phase2-security	Authentication and encryption algorithms. To specify authentication and encryption algorithms, use the following command: `Admin@nodename# create vpn client-security-profiles ... phase2-security new auth-alg <auth-alg-name> encrypt-alg <encrypt-alg-name>` Available values: • **auth-alg**: select an authentication algorithm. ◦ **MD5** ◦ **SHA1** ◦ **SHA256** ◦ **SHA384** ◦ **SHA512** • **encrypt-alg**: select an encryption algorithm. ◦ **DES** ◦ **3DES** ◦ **AES128** ◦ **AES192** ◦ **AES256**

Example of Creating and Editing a VPN Security Profile

Creating a new VPN security profile:

```
Admin@nodename# create vpn server-security-profiles <profile-name>
<parameters>
Admin@nodename# create vpn client-security-profiles <profile-name>
<parameters>
…
```

```
Admin@nodename# create vpn server-security-profiles name "New server
VPN profile"
```

Editing parameters of a VPN security profile:

```
Admin@nodename# set vpn server-security-profiles <profile-name>
<parameters>
Admin@nodename set vpn client-security-profiles <profile-name>
<parameters>
...
Admin@nodename# set vpn server-security-profiles "New server VPN
profile" phase1-security [ SHA1/AES128 SHA1/3DES ] phase2-security
[ SHA1/AES128 SHA1/3DES ]
Admin@nodename# set vpn server-security-profiles "New server VPN
profile" dh-groups [ "Group 16 Prime 4096 bit" ]
Admin@nodename# set vpn server-security-profiles "New server VPN
profile" nat-keepalive 20
```

Remove parameters of a VPN security profile:

```
Admin@nodename# delete vpn server-security-profiles <profile-name>
<parameters>
Admin@nodename# delete vpn client-security-profiles <profile-name>
<parameters>
...
Admin@nodename# delete vpn server-security-profiles "New server VPN
profile" phase1-security [ MD5/AES128 ]
Admin@nodename# delete vpn server-security-profiles "New server VPN
profile" phase2-security [ MD5/AES128 ]
Admin@nodename# delete vpn server-security-profiles "New server VPN
profile" dh-groups [ "Group 16 Prime 4096 bit" ]
```

Viewing information on configured VPN security profiles:

```
Admin@nodename# show vpn server-security-profiles <profile-name>
Admin@nodename# show vpn client-security-profiles <profile-name>
```

```
...
Admin@nodename# show vpn client-security-profiles "New client VPN
profile"
```

CONFIGURING LIBRARIES

Configuring Libraries (Description)

Configuring services

You configure this section at the **libraries services** level.

To add a new service, use the following command:

```
Admin@nodename# create libraries services <parameter>
```

Provide the following parameters:

Parameter	Description
name	Service name.
description	Service description.
protocols	Network protocol and source/destination ports: • **protocol**: network protocol. • **alg**: L7 gateway, only for TCP and UDP (SIP, H323 and FTP, TFTP for UDP protocol are supported). • **dest-ports**: destination port(s) (you can specify a port number or a port range). • **source-ports**: source port(s) (you can specify a port number or a port range).

To edit an existing service, use the following command:

```
Admin@nodename# set libraries services <service-name> <parameter>
```

To update the network protocol and source/destination ports, use the following command:

```
Admin@nodename# set libraries services <service-name> protocols
( <protocol-filter> )
```

<protocol-filter> — filter set using protocols parameter values.

Then specify the new values for the **protocols** parameter.

To add a network protocol and source/destination ports, use the following command:

```
Admin@nodename# set libraries services <service-name> protocols new
<parameter>
```

Then specify **protocol**, **dest-ports**, **source-ports**.

To delete a service, use the following command:

```
Admin@nodename# delete libraries services <service-name>
```

To delete specific network protocols from a service, use the following command:

```
Admin@nodename# delete libraries services <service-name> protocols
( <protocol-filter> )
```

To display information about all or individual services, use the following commands:

```
Admin@nodename# show libraries services
Admin@nodename# show libraries services <service-name>
```

Configuring service groups

This section is located at the **libraries service-groups** level. To create a service group, use the following command:

```
Admin@nodename# create libraries service-groups <parameter>
```

Specify the parameters:

Parameter	Description
name	Service group name.
description	Group description.
services	Services to add to the group.

To edit a services group (parameters available to update are identical to those used to create a group), use the following command:

```
 Admin@nodename# set libraries service-groups <service-group-name>
<parameter>
```

To add services to a group, use the following command:

```
 Admin@nodename# set libraries service-groups <service-group-name>
[ <service1> < service2> ... ]
```

To delete a service group or services from it, use the following commands:

```
Admin@nodename# delete libraries service-groups <service-group-name>
Admin@nodename# delete libraries service-groups <service-group-name>
services [ <service> ... ]
```

To display information about all existing lists, use the following commands:

```
 Admin@nodename# show libraries service-groups
```

To display information about a specific list, use the following command:

```
Admin@nodename# show libraries service-groups <service-group-name>
```

To display a list of services added to a group, use the following command:

```
Admin@nodename# show libraries service-groups <service-group-name>
services
```

Configuring IP addresses

This section is located at the **libraries ip-list** level.

To create an IP address group, use the following command:

```
Admin@nodename# create libraries ip-list <parameter>
```

Provide the following parameters:

Parameter	Description
name	Address list name.
description	List description.
threat-lvl	Threat level: • **very-low**: very low threat level • **low**: low threat level • **medium**: medium threat level • **high**: high threat level • **very-high**: very high threat level.
type	List type: • **local**: local • **updatable**: if the list is updatable, specify URL address for downloading updates (**url**). List update frequency is set by the **schedule** parameter in the crontab format.

Parameter	Description
	Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".
lists	Select existing IP lists to add to the list being created.
ips	IP addresses or a range of IP addresses to include in the list. Format: <ip>, <ip/mask>, or <ip_range_start-ip_range_end>.

To edit a list (parameters available to update are identical to those used to create a list), use the following command:

```
Admin@nodename# set libraries ip-list <ip-list-name> <parameter>
```

To add new addresses to a list, use the following command:

```
Admin@nodename# set libraries ip-list <ip-list-name> [ <ip1>
<ip2> ... ]
```

To delete an entire address list or individual IP addresses it contains, use the following commands:

```
Admin@nodename# delete libraries ip-list <ip-list-name>
Admin@nodename# delete libraries ip-list <ip-list-name> ips [ <ip1>
<ip2>... ]
```

To display information about all existing lists, use the following command:

```
Admin@nodename# show libraries ip-list
```

To display information about an individual list, specify the IP address list name:

```
Admin@nodename# show libraries ip-list <ip-list-name>
```

To display the contents of an IP address list, use the following command:

```
Admin@nodename# show libraries ip-list <ip-list-name> items
```

Configuring URL lists

You configure URL lists at the **libraries url-list** level.

To add a new URL list, use the following command:

```
Admin@nodename# create libraries url-list <parameter>
```

Specify the following parameters:

Parameter	Description
name	URL list name.
description	URL list description.
type	List type: • **local**: local • **updatable**: if the list is updatable, specify URL address for downloading updates (**url**). List update frequency is set by the **schedule** parameter in the crontab format. Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7.

Parameter	Description
	• Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".
urls	URLs to add to the list.
case-sensitivity	Case sensitivity in URL writing: • sensitive: sensitive to the case of letters in the address • insensitive: insensitive to the case of letters in the address • domain: list of domain addresses

Example command to edit a URL list:

```
Admin@nodename# set libraries url-list <url-list-name> <parameter>
```

The parameters for which values are available to update are listed in the table above.

To delete an entire URL list or individual URLs from it, use the following commands:

```
Admin@nodename# delete libraries url-list <url-list-name>
Admin@nodename# delete libraries url-list <url-list-name> urls
[ <url> ... ]
```

To display information about all URL lists, a specific URL list, or about the addresses from a specific list, use the following commands:

```
Admin@nodename# show libraries url-list
Admin@nodename# show libraries url-list <url-list-name>
Admin@nodename# show libraries url-list <url-list-name> urls
```

Configuring time sets

This section is located at the **libraries time-sets** level.

To create a group, use the following command:

```
Admin@nodename# create libraries time-sets <parameter>
```

Provide the following parameters:

Parameter	Description
name	Group name.
description	Group description.
time-set	• **interval-name**: repetition interval name. • **type**: repetition interval type: ◦ **daily**: daily: ■ **time-from**: start time (format: HH:MM). ■ **time-to**: end time (format: HH:MM). ■ **all-day on**: all day. ◦ **weekly**: every week: ■ **time-from**: start time (format: HH:MM). ■ **time-to**: end time (format: HH:MM). ■ **all-day on**: all day. ■ **days [Mon | Tue | Wed | Thu | Fri | Sat | Sun]**: days of the week. ◦ **monthly**: every month: ■ **time-from**: start time (format: HH:MM). ■ **time-to**: end time (format: HH:MM). ■ **all-day on**: all day. ■ **days**: days of the month from 1 to 31. ◦ **fixed**: one time: ■ **time-from**: start time (format: HH:MM). ■ **time-to**: end time (format: HH:MM). ■ **all-day on**: all day. ■ **fixed-date**: desired date (format: YYYY-MM-DD). ◦ **span**: repeating events: ■ **time-from**: start time (format: HH:MM). ■ **time-to**: end time (format: HH:MM). ■ **all-day on**: all day. ■ **fixed-date-from**: start date (format: YYYY-MM-DD).

Parameter	Description
	■ **fixed-date-to**: end date (format: YYYY-MM-DD). ◦ **range**: date range: ■ **time-from-enabled <on | off>**: enable/disable setting the interval start date. ■ **fixed-date-from**: start date (format: YYYY-MM-DD). ■ **time-from**: start time (format: HH:MM). ■ **time-to-enabled <on | off>**: enable/disable setting the interval end date. ■ **fixed-date-to**: end date (format: YYYY-MM-DD). ■ **time-to**: end time (format: HH:MM).

To edit a time set, use the following command:

```
Admin@nodename# set libraries time-sets <time-sets-name> <parameter>
```

The parameters available to update are listed in the table above.

To edit an interval specified for a time set, use the following command:

```
Admin@nodename# set libraries time-sets <time-sets-name> ... time-set
<time-set-type> ( <time-set-filter> )
```

The new values are then specified as follows; <time-set-filter> — filter for interval current values.

To add a new item to an existing group, use the following command:

```
Admin@nodename# create libraries time-sets <time-sets-name> ... time-
set <time-set-type> new
```

To delete a group of items, use the following command:

```
Admin@nodename# delete libraries time-sets <time-sets-name>
```

To delete an item from a time set, use the following command:

```
Admin@nodename# delete libraries time-sets <time-sets-name> <time-set-type> ( <time-set-filter> )
```

To display information about all time sets, use the following command:

```
Admin@nodename# show libraries time-sets
```

To display information about an individual time set, use the following command:

```
Admin@nodename# show libraries time-sets <time-sets-name>
```

To display information about group items with the same repeat type, use the following command:

```
Admin@nodename# show libraries time-sets <time-sets-name> <time-set-type>
```

Configuring bandwidth pools

This section is located at the **libraries bandwidth-pools** level.

To add a new bandwidth pool, use the following command:

```
Admin@nodename# create libraries bandwidth-pools <parameter>
```

Provide the following parameters:

Parameter	Description
name	Bandwidth pool name.
description	Bandwidth pool description.
rate	Data transfer rate (in kbit/s).

Parameter	Description
dscp	DSCP value for QoS (if set, it will be added to each IP packet; available values: from 0 to 63).

To edit bandwidth pool parameters, use the following command:

```
Admin@nodename# set libraries bandwidth-pools <bandwidth-name>
<parameter>
```

You can change the same parameters as those for creating a bandwidth pool.

To delete a bandwidth pool, use the following command:

```
Admin@nodename# delete libraries bandwidth-pools <bandwidth-name>
```

Specify **show** to display information on all bandwidth pools:

```
Admin@nodename# show libraries bandwidth-pools
```

or about an individual pool:

```
Admin@nodename# show libraries bandwidth-pools <bandwidth-name>
```

Configuring response pages

This section is located at the **libraries response-pages** level. You can create the following response pages (<response-page-type>):

- **blockpage**: block page
- **captiveportal-user-auth**: captive portal authorization page
- **captiveportal-user-session**: captive portal user session
- **content-warning**: content warning page
- **ftp-client**: FTP over HTTP view page

- **proxy-portal**: web portal page
- **pp-login-ssh**: web portal login page for SSH
- **pp-login-rdp**: web portal login page for RDP
- **totp-init-page**: TOTP initialization page

To create a response page, use the following command:

```
Admin@nodename# create libraries response-pages <response-page-type>
<parameter>
```

Specify the following parameters:

Parameter	Description
name	Response page name.
description	Response page description.
original-template	Select a basic response page. Basic response pages for a block page (**blockpage**): • **blockpage_en**: block response page in English. • **blockpage_ru**: block response page in Russian. Basic response pages for a Captive portal authentication page (**captiveportal-user-auth**): • **captiveportal_user_auth_en**: response page for user authentication using the captive portal in English. • **captiveportal_user_auth_ru**: response page for user authentication using the captive portal in Russian. • **captiveportal_user_auth_policy_en**: response page for user authentication using the captive portal in English. In addition to the authentication form, the response page displays network usage rules (usage agreement) and requires the user to accept the access policy rules. • **captiveportal_user_auth_policy_ru**: response page for user authentication using the captive portal in Russian. In addition to the authentication form, the response page displays network usage rules (usage agreement) and requires the user to accept the access policy rules. • **captiveportal_user_auth_email_en**: response page for user authentication using the captive portal in English which allows users to register in the system using email confirmation.

Parameter	Description
	• **captiveportal_user_auth_email_ru**: response page for user authentication using the captive portal in Russian which allows users to register in the system using email confirmation. • **captiveportal_user_auth_sms_en**: response page for user authentication using the captive portal in English which allows users to register in the system using SMS confirmation. • **captiveportal_user_auth_sms_ru**: response page for user authentication using the captive portal in Russian which allows users to register in the system using SMS confirmation. • **captiveportal_user_policy_en**: response page for user authentication using the captive portal in English. The templates do not require the name and password entry; they just display the network terms of use (user agreement) and require the user to agree to the access policy. For this response page, you need to specify **Policy accept** as the authentication method in the auth profile. • **captiveportal_user_policy_ru**: response page for user authentication using the captive portal in Russian. The templates do not require the name and password entry; they just display the network terms of use (user agreement) and require the user to agree to the access policy. For this response page, you need to specify **Policy accept** as the authentication method in the auth profile. Basic response pages for the captive portal user session page (**captiveportal-user-session**): • **captiveportal_user_session_en**: template in English which allows the user to finish his/her authenticated session by going to http://logout.captive or http://USERGATE_IP/cps. • **captiveportal_user_session_ru**: template in Russian which allows the user to finish his/her authenticated session by going to http://logout.captive or http://USERGATE_IP/cps. Basic response pages for a warning page (**content-warning**): • **content_warning_en**: warning page template in English displayed when a content filtering rule with **Warn** action is triggered. • **content_warning_ru**: warning response page in Russian that is displayed if a content filtering rule with the **Warn** action is triggered.

Parameter	Description
	Basic response pages for the FTP over HTTP view page (**ftp-client**): • **ftp_client_en**: response page in English to display FTP server content over HTTP. • **ftp_client_ru**: response page in Russian to display FTP server content over HTTP. Basic response pages for the web portal page (**proxy-portal**): • **proxy_portal_en**: response page in English to display the web portal page. • **proxy_portal_ru**: response page in Russian to display the web portal page. Basic response pages for a web portal login page for SSH (**pp-login-ssh**): • **pp_login_ssh_en**: response page in English to display an authentication page when connecting to SSH resources via the web portal. • **pp_login_ssh_ru**: response page in Russian to display an authentication page when connecting to SSH resources via the web portal. Basic response pages for a web portal login page for RDP (**pp-login-rdp**): • **pp_login_rdp_en**: response page in English to display an authentication page when connecting to RDP resources via the web portal. • **pp_login_rdp_ru**: response page in Russian to display an authentication page when connecting to RDP resources via the web portal. Basic response pages for TOTP initialization page (**totp-init-page**): • **totp_init_page_en**: response page in English to display a MFA TOTP device initialization page for VPN users. • **totp_init_page_ru**: response page in Russian to display a MFA TOTP device initialization page for VPN users.
default	Use the default response page: • **on** • **off**

To edit template values, use the following command:

```
 Admin@nodename# set libraries response-pages <response-page-type>
<response-page-name> <parameters>
```

Then specify the parameters listed in the table above.

To delete a response page, use the following command:

```
 Admin@nodename# delete libraries response-pages <response-page-type>
<response-page-name>
```

To display information about all existing response pages, use the following command:

```
 Admin@nodename# show libraries response-pages
```

To display information about an individual response page type, use the following command:

```
 Admin@nodename# show libraries response-pages <response-page-type>
```

To display information about an individual response page, use the following command:

```
 Admin@nodename# show libraries response-pages type <response-page-
type> <response-page-name>
```

Configuring Application Groups

This section is configured at the **libraries application-groups** level.

To create an application group, use the following command:

```
Admin@nodename# create libraries application-groups
```

Specify the parameters:

Parameter	Description
name	Application group name.
description	Application group description.
apps	Applications to add to the group.

To edit parameters, use the following command:

```
Admin@nodename# set libraries application-groups <app-group-name>
```

To add new applications to an existing group, use the following command:

```
Admin@nodename# set libraries application-groups <app-group-name> apps
[ <application> ... ]
```

To delete an entire application group or individual applications from it, use the following commands:

```
Admin@nodename# delete libraries application-groups <app-group-name>
Admin@nodename# delete libraries application-groups <app-group-name>
apps [ <application> ... ]
```

To display information about all existing application groups, use the following commands:

```
Admin@nodename# show libraries application-groups
```

To display information about an individual URL category group, use the following command:

```
Admin@nodename# show libraries application-groups <app-group-name>
```

To display applications from a specific list, use the following command:

```
 Admin@nodename# show libraries application-groups <app-group-name>
apps
```

Configuring Application Profiles

Application profiles are configured at the **libraries application-profile** level.

To create an application profile, use the following command:

```
Admin@nodename# create libraries application-profile <parameter>
```

Provide the following parameters:

Parameter	Description
name	Application profile name.
description	Application profile description.
filters	Filters for selecting relevant signatures from application signature library.
unknown-application-settings	Signature options for unknown applications.

To edit an existing applications profile, use the following command:

```
 Admin@nodename# set libraries application-profile <application-
profile-name> <parameter>
```

The parameters available to update are identical to those used to create a profile.

To view information on all application profiles, use the following command:

```
Admin@nodename# show libraries application-profile
```

To display information about an individual IPS profile, use the following command:

```
Admin@nodename# show libraries application-profile <application-
profile-name>
```

Example of creating an application profile:

```
Admin@nodename# create libraries application-profile name "Test app
profile 1" description "Test app profile 1 description" filters new
action drop value "category = Games" enabled on
Admin@nodename# show libraries application-profile "Test app profile 1"

name           : Test app profile 1
description    : Test app profile 1 description
filters        :
    enabled           : on
    value             : category = Games
    enable-setting    : on
    action            : drop
    pcap              : off
    track-by          : src
    duration          : 0 days 0 hours 5 minutes
```

To remove an application profile, use the following command:

```
Admin@nodename# delete libraries application-profile <application-
profile-name>
```

Configuring Application Signatures

At the **libraries application-signature** level it is possible to create and configure user custom application signatures.

To create a custom application signature, use the following command:

```
Admin@nodename# create libraries application-signature <parameters>
```

Provide the following parameters:

Parameter	Description
name	The name of the signature. Cannot be modified for signatures created by UserGate.
description	Signature description. Cannot be modified for signatures created by UserGate.
signature-id	Signature group ID. Cannot be modified for signatures created by UserGate.
enabled	Signature state indicator. • **on**: enable • **off**: disable
categories	A signature category is a group of signatures that have common parameters. The list of categories can be extended. • Media streaming • Email • Coin Miners • TunnelingGames • Remote access • Conferencing • Trojan Horses • Business • Mobile • Proxies and anonymizers • Standard networks • VOIP • Web posting • Software update • File storage and backup • Web browsing • File sharing P2P • Instant messaging • Social networking
threat	Threat level defined by the signature. The following values are defined: • **very-low** • **low**

Parameter	Description
	• **medium** • **high** • **very-high**
technology	Application technology. • **browser-based**: browser-based web application • **client-server**: client-server application • **network-protocol**: network protocol • **peer-to-peer**: peer-to-peer application
type	Signature type: • **app**: application signature • **proto**: protocol signature • **support**: supplementary signature
uasl	Application signature description using UASL syntax.

To edit a previously created application signature, use the following command:

```
Admin@nodename# set libraries application-signature <application-
signature-name> <parameters>
```

Parameters which could be updated are the same parameters which are available when creating a signature.

To view information on all application signatures, use the following command:

```
Admin@nodename# show libraries application-signature
```

To view information on a specific signature, use the following command:

```
Admin@nodename# show libraries application-signature <application-
signature-name>
```

Example of creating an application signature:

```
Admin@nodename# create libraries application-signature name "Test app
signature 2" description "Test app signature 2 description" categories
[ "Web browsing" ] signature-id 2 technology browser-based threat low
type app uasl "UASL(.dst_addr=192.168.10.1;)"
Admin@nodename# show libraries application-signature "Test app
signature 2"

signature-id    : 2
name            : Test app signature 2
threat          : low
technology      : browser-based
categories      : Web browsing
uasl            : UASL(.dst_addr=192.168.10.1;)
owner           : You
type            : custom
description     : Test app signature 2 description
```

To remove a previously created application signature, use the following command:

```
Admin@nodename# delete libraries application-signature <application-
signature-name>
```

Configuring email addresses

This section is located at the **libraries email-list** level.

To add a new email group, use the following command:

```
Admin@nodename# create libraries email-list <parameter>
```

Specify the parameters:

Parameter	Description
name	Email group name.
description	Email group description.

Parameter	Description
type	List type: • **local**: local • **updatable**: if the list is updatable, specify URL address for downloading updates (**url**). List update frequency is set by the **schedule** parameter in the crontab format. Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".
emails	Emails to add to the group.

To edit information about an email group, use the following command:

```
Admin@nodename# set libraries email-list <email-list-name> <parameter>
```

The parameters available to update are the same as those for creating an email group.

To delete a group or individual emails from it, use the following commands:

```
Admin@nodename# delete libraries email-list <email-list-name>
Admin@nodename# delete libraries email-list <email-list-name> emails
[ <email> ... ]
```

To view information about all existing groups, about individual groups, or about emails in a group, use the following commands:

```
Admin@nodename# show libraries email-list
Admin@nodename# show libraries email-list <email-list-name>
Admin@nodename# show libraries email-list <email-list-name> emails
```

Configuring phones

The **Phones** section is configured at the **libraries phone-list** level.

To create a phone group, use the following command:

```
Admin@nodename# create libraries phone-list <parameter>
```

Provide the following parameters:

Parameter	Description
name	Phone group name.
description	Phone group description.
type	List type: • **local**: local • **updatable**: if the list is updatable, specify URL address for downloading updates (**url**). List update frequency is set by the **schedule** parameter in the crontab format. Crontab format: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6; where 0 is Sunday). You can set each field as follows: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".
phones	Phones to add to the group.

To edit information about a phone group, use the following command:

```
Admin@nodename# set libraries phone-list <phone-list-name> <parameter>
```

The parameters available to update are listed in the table above.

To delete a group or individual phones from it, use the following commands:

```
Admin@nodename# delete libraries phone-list <phone-list-name>
Admin@nodename# delete libraries phone-list <phone-list-name> phones
[ <phone> ... ]
```

To view information about all existing groups, use the following command:

```
Admin@nodename# show libraries phone-list
```

To view information about an individual phone group, use the following command:

```
Admin@nodename# show libraries phone-list <phone-list-name>
```

To display phones included in a group, use the following command:

```
Admin@nodename# show libraries phone-list <phone-list-name> phones
```

Configuring IDPS Signatures

At the **libraries ips-signature** level it is possible to create and configure user IDPS signatures.

To create a custom IDPS signature, use the following command:

```
Admin@nodename# create libraries ips-signature <parameters>
```

Provide the following parameters:

Parameter	Description
name	The IDPS signature name. Cannot be modified for signatures created by UserGate.
description	The IDPS signature description. Cannot be modified for signatures created by UserGate.
signature-id	Signature group ID. Cannot be modified for signatures created by UserGate.
enabled	Signature state indicator. • **on**: enable • **off**: disable
threat	Threat level defined by the signature. The following values are defined: • **very-low** • **low** • **medium** • **high** • **very-high** Cannot be modified for signatures created by UserGate.
action	Responsive action to signature triggering. The following values are defined: • **none**: action is not defined • **pass**: skip the packet • **drop**: drop the packet • **rst**: drop the packet and close TCP connection (sending TCP reset) • **block**: block source and/or destination IP address.
log	Logging: • **on**: enable logging • **off**: disable logging.
os	Operating system type for which the signature is defined: • **windows** • **linux**

Parameter	Description
	• **bsd** • **macos** • **solaris** • **cisco** • **ios** • **android** • **other** Cannot be modified for signatures created by UserGate.
pcap	Tracking signature triggering and logging it to PCAP file. • **on**: enable • **off**: disable
track-by	Applying **block** or **rst** actions in response to signature triggering: • **src**: the **block** or **rst** actions are applied to the source IP address of the packet. • **dst**: the **block** or **rst** actions are applied to the destination IP address of the packet. • **both**: the **block** or **rst** action is applied to both the source and destination IP addresses of the packet.
duration	Blocking duration for **block** action.
uasl	Description of the signature using the UASL syntax. Cannot be modified for signatures created by UserGate.
cve	Vulnerability ID according to CVE registry.
bdu	Vulnerability ID according to the BDU registry.
url	Optional link to a resource with the description of the vulnerability.
category	A signature category is a group of signatures that have common parameters. The list of categories (can be extended): • adware pup • attack_response: signatures that specify responses to known network attacks. • coinminer: downloading, installation, and runtime activity of known miners.

Parameter	Description
	• dns: known DNS vulnerabilities • dos: known signatures of denial-of-service (DoS) attacks • exploit: known exploit signatures. • ftp: known FTP vulnerabilities. • imap: known IMAP vulnerabilities. • info: potential data leak. • ldap: known LDAP vulnerabilities. • malware: downloading, installation, and runtime activity of known malware. • misc: other known signatures. • netbios: known NetBIOS protocol vulnerabilities. • phishing: known phishing attack signatures. • pop3: known POP3 protocol vulnerabilities. • rpc: known RPC protocol vulnerabilities. • scada: known SCADA protocol vulnerabilities. • scan: signatures of attempts to scan the network for known applications. • shellcode: signatures specifying known attempts at launching shells. • smtp: known SMTP protocol vulnerabilities. • snmp: known SNMP protocol vulnerabilities. • sql: known SQL vulnerabilities. • telnet: known attempts at cracking via the telnet protocol. • tftp: known TFTP protocol vulnerabilities. • user_agents: signatures of suspicious Useragents. • voip: known VoIP protocol vulnerabilities. • web_client: signatures of known attempts at cracking various web clients, such as Adobe Flash Player. • web_server: signatures specifying known attempts at cracking various web servers. • web_specific_apps: signatures specifying known attempts at cracking various web applications. • worm: signatures specifying the network activity of known network worms. Cannot be modified for signatures created by UserGate.
classtype	The signature class determines the attack type that is detected using this signature. In addition, it determines the general events that are not related o the attack but can be relevant in

Parameter	Description
	certain cases; e.g., detecting the establishment of a TCP session. The class list (can be extended): • arbitrary-code-execution: attempt to run arbitrary code. • attempted-admin: attempt to obtain administrative privileges. • attempted-dos: attempt to launch a Denial-of-Service (DoS) attack. • attempted-recon: attempt to launch an attack aimed at leaking data. • attempted-user: attempt to obtain user privileges. • bad-unknown: potentially unwanted traffic. • command-and-control: attempt to communicate to C&C center. • default-login-attempt: attempt to log in with the default username/password. • denial-of-service — Denial of Service attack detected. • exploit-kit: exploit kit detected. • misc-activity: other activity. • misc-attack — attack detected. • shellcode-detect: shell code detected. • string-detect: suspicious string detected. • suspicious-login: attempt to log in using suspicious user name. • trojan-activity: network trojan detected. • web-application-attack — web application attack detected. Cannot be modified for signatures created by UserGate.

To edit a previously created IDPS signature, use the following command:

```
Admin@nodename# set libraries ips-signature <ips-signature-name>
<parameters>
```

Parameters which could be updated are the same parameters which are available when creating a signature.

To view information on all IDPS signatures, use the following command:

```
Admin@nodename# show libraries ips-signature
```

To view information on a specific signature, use the following command:

```
Admin@nodename# show libraries ips-signature <ips-signature-name>
```

Example of creating an IDPS signature:

```
Admin@nodename# create libraries ips-signature name "Test signature"
action none threat low description "Test signature description" log on
pcap on url example.org uasl "UASL(.name=\"EXAMPLE\";)" enabled off
Admin@nodename# show libraries ips-signature "Test signature"

signature-id    : 5
name            : Test signature
enabled         : off
description     : Test signature description
threat          : low
action          : none
log             : on
pcap            : on
track-by        : src
duration        : 0 days 0 hours 5 minutes
uasl            : UASL(.name="EXAMPLE";)
url             : example.org
owner           : You
type            : custom
```

To remove a previously created IDPS signature, use the following command:

```
Admin@nodename# delete libraries ips-signature <ips-signature-name>
```

Configuring IPS profiles

IDPS profiles are configured at the **libraries ips-profile** level.

To create a profile for the intrusion detection system, use the following command:

```
Admin@nodename# create libraries ips-profile <parameter>
```

Provide the following parameters:

Parameter	Description
name	IPS profile name.
description	IPS profile description.
filters	Filters for selecting relevant signatures from IDPS signature library.

To edit an existing IDPS profile, use the following command:

```
 Admin@nodename# set libraries ips-profile <ips-profile-name>
<parameter>
```

Using the following command, you can reconfigure the settings of the IDPS system signatures included in the rule:

```
Admin@nodename# set libraries ips-profile <ips-profile-name> override
signature <signature-name> <parameters>
```

The following command allows you to return previously reconfigured parameters of the IDPS system signature to their original value:

```
Admin@nodename# set libraries ips-profile <ips-profile-name> override
signature <signature-name> restore-default
```

To view information on all IDPS profiles, use the following command:

```
Admin@nodename# show libraries ips-profile
```

To view information on a specific profile, use the following command:

```
Admin@nodename# show libraries ips-profile <ips-profile-name>
```

Example of creating an IDPS profile:

```
Admin@nodename# create libraries ips-profile name testipsprofile1
filters new enabled on value "threat > 2 AND owner = 'UserGate'"
Admin@nodename# show libraries ips-profile testipsprofile1

name           : testipsprofile1
filters        :
    enabled    : on
    value      : threat > 2 AND owner = 'UserGate'
```

To remove an IDPS profile, use the following command:

```
Admin@nodename# delete libraries ips-profile <ips-profile-name>
```

Configuring notification profiles

You configure notification profiles for SMTP (via email) and SMPP (via SMS) at the **libraries notification-profiles** level.

To add a new SMTP notification profile:

```
Admin@nodename# create libraries notification-profiles smtp <parameter>
```

Specify the following parameters:

Parameter	Description
name	Profile name.
description	Profile description.
host	The IP address or FQDN of the SMTP server that will be used for sending emails.
port	The TCP port used by the SMTP server. Usually, SMTP uses port 25, and SMTP with SSL uses port 465. Consult your email server administrator regarding this value.

Parameter	Description
connection-security	The following outgoing email security options are available: • **none**. • **starttls**. • **ssl**.
authentication	Enable/disable authorization when connecting to the SMTP server: • **on** • **off**
login	Login name to connect to the SMTP server.
password	Password to connect to the SMTP server.

To create an SMS (SMPP) notification profile, use the following command:

```
 Admin@nodename# create libraries notification-profiles smpp
<parameter>
```

Provide the following parameters:

Parameter	Description
name	Profile name.
description	Profile description.
host	IP address or FQDN of an SMPP server to use to send SMS.
port	TCP port to use to connect to the SMPP server. Usually, the port used for the SMPP protocol is 2775, when using SSL — 3550.
ssl	Enable/disable SSL encryption: • **on** • **off**
login	The account name for connecting to the SMPP server.
password	The account password for connecting to the SMPP server.

Parameter	Description
phone-translation-rules	Phone translation rules. These rules are used to ensure that the provider requirements are met. For example, to replace all numbers starting with +7 to 8, use the following command: `Admin@nodename# set libraries notification-profiles smpp <profile-name> phone-translation-rules + [ +7¦8 ]`
source-ton	Type of number for the event source: • **0**: unknown • **1**: international • **2**: national • **3**: network specific • **4**: subscriber number • **5**: alphanumeric • **6**: abbreviated.
dest-ton	Type of number for destination: • **0**: unknown • **1**: international • **2**: national • **3**: network specific • **4**: subscriber number • **5**: alphanumeric • **6**: abbreviated.
source-npi	Numbering Plan Indicator for the source: • **0**: Unknown. • **1**: ISDN/telephone numbering plan (E.163/E.164) • **3**: data numbering plan (X.121) • **4**: telex numbering plan (F.69) • **6**: land Mobile (E.212) • **8**: national numbering plan • **9**: private numbering plan • **10**: ERMES numbering plan (ETSI DE/PS 3 01-3) • **13**: Internet (IP).

Parameter	Description
	• **18**: WAP Client Id (to be defined by WAP Forum).
dest-npi	Numbering Plan Indicator for the destination: • **0**: Unknown. • **1**: ISDN/telephone numbering plan (E.163/E.164) • **3**: data numbering plan (X.121) • **4**: telex numbering plan (F.69) • **6**: land Mobile (E.212) • **8**: national numbering plan • **9**: private numbering plan • **10**: ERMES numbering plan (ETSI DE/PS 3 01-3) • **13**: Internet (IP). • **18**: WAP Client Id (to be defined by WAP Forum).

To edit a notification profile, use the following command:

```
 Admin@nodename# set libraries notification-profiles <smtp | smpp>
<profile-name> <parameter>
```

SMTP and SMPP profile parameters available to change are listed in the respective tables above.

To delete a profile, use the following command:

```
 Admin@nodename# delete libraries notification-profiles <smtp | smpp>
<profile-name>
```

You can also delete phone translation rules from SMPP notifications:

```
 Admin@nodename# delete libraries notification-profiles smpp <profile-
name> phone-translation-rules [ phone1¦phone2 ]
```

To display information about all existing notification profiles, use the following command:

```
Admin@nodename# show libraries notification-profiles
```

To display information about all notification profiles of a specific type, use the following command:

```
Admin@nodename# show libraries notification-profiles <smtp | smpp>
```

To display information about an individual notification profile, use the following command:

```
Admin@nodename# show libraries notification-profiles <smtp | smpp>
<profile-name>
```

Configuring Netflow profiles

This section is located at the **libraries netflow-profiles** level.

To create a Netflow profile, use the following command:

```
Admin@nodename# create libraries netflow-profiles <parameter>
```

Provide the following profile parameters:

Parameter	Description
name	Netflow profile name.
description	Profile description.
ip	IP address of a Netflow collector to which the sensor will send the statistics.
port	UDP port on which the Netflow collector will receive the statistics.

Parameter	Description
protocol	Netflow protocol version to use (it should be identical on the sensor and the collector): • **5**: Netflow, version 5. • **9**: Netflow, version 9. • **10**: Netflow, version 10.
active-timeout	Time after which statistics will be sent to the collector without waiting for the flow to finish (e.g., transferring a large file over the network). In seconds. Default value: 1800 seconds; maximum value: 3600 seconds.
inactive-timeout	Time allowed for termination of an inactive flow (in seconds). Default value — 15 seconds; maximum value — 3600 seconds.
max-flows	Maximum number of counted flows from which statistics are gathered and sent. After the specified number of flows is reached, all subsequent flows will not be counted (this limitation is necessary to ensure protection against DoS attacks); default value — 2000000; to remove the limitation, set this parameter to 0.
nat-events	Enable/disable sending information about NAT conversions to Netflow statistics: • **on** • **off**
refresh-rate	Number of packets; after it is reached the template is sent to the receiving host (only for NetFlow 9/10). The template contains information about the configuration of the device and various statistical information. The default value is 20 packets.
timeout-rate	Time after which the old template is sent to the receiving host (Netflow 9/10 versions only). The template contains information about the configuration of the device and various statistical information. The default value is 1800 seconds.

To edit an existing profile, use the following command:

```
Admin@nodename# set libraries netflow-profiles <profile-name>
```

The parameters you can change the values of are listed in the table above.

To delete a Netflow profile, use the following command:

```
Admin@nodename# delete libraries netflow-profiles <profile-name>
```

To display information about all or individual Netflow profiles, use the following commands:

```
Admin@nodename# show libraries netflow-profiles
Admin@nodename# show libraries netflow-profiles <profile-name>
```

Configuring LLDP profiles

You create and configure LLDP (Link Layer Discovery Protocol) profile properties at the **libraries lldp-profiles** level.

To create an LLDP profile, use the following command:

```
Admin@nodename# create libraries lldp-profiles <parameter>
```

Specify the following parameters:

Parameter	Description
name	The name of the LLDP profile.
description	Profile description.
port-status	Modes: • **rx**: only receive LLDP data. UserGate will not send any LLDP information but will analyze that received from its neighbors. • **tx**: only send LLDP data. UserGate will send LLDP information but will discard that received from its neighbors. • **rx-tx**: receive and send LLDP data. UserGate will send LLDP information and analyze that received from its neighbors.

To edit profile information, use the following command:

```
Admin@nodename# set libraries lldp-profiles <profile-name> <parameter>
```

The parameters available to update are identical to those used to create a profile.

To delete an LLDP profile, use the following command:

```
Admin@nodename# delete libraries lldp-profiles <profile-name>
```

To display profile information, use the following command:

```
Admin@nodename# show libraries lldp-profiles
Admin@nodename# show libraries lldp-profiles <profile-name>
```

Configuring SSL profiles

You configure SSL profiles at the **libraries ssl-profiles** level.

To create an SSL profile, use the following command:

```
Admin@nodename# create libraries ssl-profiles <parameter>
```

Specify the following parameters:

Parameter	Description
name	The name of the SSL profile.
description	Profile description.
min-tls-version	Minimum TLS version that can be used in this profile: • **tls1**. • **tls1.1**. • **tls1.2**.

Parameter	Description
max-tls-version	Maximum TLS version that can be used in this profile: • **tls1**. • **tls1.1**. • **tls1.2**. • **tls1.3**.
ssl-ciphers	Select the necessary digital signature and encryption algorithms.
ssl-ciphers-suite	Set encryption algorithms for standard protocols. This parameter is used to select the required signature and encryption algorithms for standard TLS protocols. Specify a version: • **tls1**. • **tls1.1**. • **tls1.2**. • **tls1.3**.

To edit profile information, use the following command:

```
Admin@nodename# set libraries ssl-profiles <profile-name> <parameter>
```

The parameters available to update are identical to those used to create a profile.

To delete an entire SSL profile or individual digital signature and encryption algorithms from it, use the following commands:

```
Admin@nodename# delete libraries ssl-profiles <profile-name>
Admin@nodename# delete libraries ssl-profiles <profile-name> ssl-
ciphers [ cipher ... ]
```

To display information about SSL profiles, use the following command:

```
Admin@nodename# show libraries ssl-profiles
Admin@nodename# show libraries ssl-profiles <profile-name>
```

Configuring BFD profiles

This section is located at the **libraries bfd-profile** level

To create a BFD profile, use the following command:

```
Admin@nodename# create libraries bfd-profile
```

Specify the following parameters:

Parameter	Description
name	The name of the BFD profile.
description	Profile description.
detect-multiplier	Detection time multiplier; affects detection time for connection problems.
receive-interval	BFD control packet reception interval (minimum time required between packets); specified in milliseconds.
transmit-interval	BFD control packet transmission interval; interval should be agreed between nodes; specified in milliseconds.
echo-receive-interval	Minimum interval for echo packets reception by this system; specified in milliseconds.
echo-transmit-interval	Minimum interval for BFD echo packets transmission by this system; specified in milliseconds.
echo-mode	Enable/disable **Echo mode** for data transmission.
pessive-mode	Enable or disable the **Passive** mode.
ttl	Minimum expected TTL for an incoming BFD control packet. Can take values from 1 to 254.

To update parameters of a BFD profile, use the following command:

```
Admin@nodename# set libraries bfd-profile <profile-name> <parameter>
```

To delete a profile, use the following command:

```
Admin@nodename# delete libraries bfd-profile <profile-name>
```

To display information on all BFD profiles or on a certain BFD profile, use the following commands:

```
Admin@nodename# show libraries bfd-profile
Admin@nodename# show libraries bfd-profile <profile-name>
```

Configuring UserID Agent Syslog Filters

Syslog filters are created and configured at the **libraries syslog-filters** level.

To create a syslog filter, use the following command:

```
Admin@nodename# create libraries syslog-filters <parameters>
```

Specify the following parameters:

Parameter	Description
name	Filter name.
description	Filter description.
login-address	String used to look up user IP address in syslog message.
login-event	String used to look up user login event in syslog message.
login-username	String used to look up username in syslog message.
logout-address	String used to look up user IP address in syslog message.
logout-event	String used to look up user logout event in syslog message.
logout-username	String used to look up username in syslog message.

To edit information on a syslog filter, use the following command:

```
Admin@nodename# set libraries syslog-filters <filter-name> <parameters>
```

Parameters which could be updated are the same parameters which are specified when creating a filter.

To display information on a syslog filter, use the following command:

```
Admin@nodename# show libraries syslog-filters <filter-name>
```

To remove a syslog filter, use the following command:

```
Admin@nodename# delete libraries syslog-filters <filter-name>
```

SETTING UP THE LOGS AND REPORTS SECTION

Configuring Log Export

The logs export feature allows you to upload information to external servers for later analysis or SIEM processing.

To create a new log export rule, use the command:

```
Admin@nodename# create logs logs-export <parameters>
```

Available parameters:

Parameter	Description
enabled	Enable/disable a rule: • **on** • **off**
name	The name of the rule.
description	A description of the rule.

Parameter	Description
server-type	Server type: • **ssh** • **ftp** • **syslog** When selecting a server type, the following additional settings are available: • **port**: the server port to which data should be sent. • **login**: account name to connect to the remote server (does not apply to syslog send method). • **password**: password of the account used to connect to the remote server (does not apply to the syslog sending method). • **path**: directory on the server to copy log files to (does not apply to the syslog sending method). • **passive**: passive FTP mode. • **transport**: for syslog server type only. TCP or UDP. • **protocol**: for syslog server type only. RFC5424 or BSD syslog RFC 3164. Select the protocol compatible with your SIEM system. • **severity**: for syslog server type only. Severity. The following values are possible: **alert**, **critical**, **error**, **warning**, **notice**, **info**. • **facility**: for syslog server type only. Object. Possible values are: **user-level**, **system-daemons**, **security-auth**, **l og-audit**,**log-alert**, **local-** (0-7). • **hostname**: for syslog server type only. A unique host name identifying the server that sends data to the Syslog server in the FQDN (Fully Qualified Domain Name) format. • **app-name**: for syslog server type only. Unique name of the application that sends data to the Syslog server.
target	IP address or domain name of the server.
logs	Logs to export: • **dns**: DNS log. • **events**: event log. • **webaccess**: web access log. • **idps**: IDPS log. • **mailsecurity**: mail traffic log. • **ssh**: SSH inspection log.

Parameter	Description
	• **traffic**: traffic log. • **userid**: UserID log. For each log, you can specify the export syntax: • **cef** • **cef-compact** • **json** • **cee-json** • **off**
schedule	Select schedule for sending logs. Not applicable to the Syslog export method. A rontab-like format in which a line appears as six fields separated by spaces. The fields specify the time as follows: (minutes: 0-59) (hours: 0-23) (days of the month: 1-31) (month: 1-12) (days of the week: 0-6, where 0 is Sunday). Each of the first five fields can be defined using: • An asterisk (*) denotes the entire range (from the first number to the last). • A dash (-) denotes a number range. For example, "5-7" means 5, 6, and 7. • Lists: comma-separated numbers or ranges. For example, "1,5,10,11" or "1-11,19-23". • An asterisk or range spacing: used for spacing out values in ranges. The increment is given after a slash. Examples: "2-10/2" means "2,4,6,8,10" while "*/2" in the "hours" field means "every two hours".

To edit the previously created rules, use the following command:

```
Admin@nodename# set logs logs-export <log-export-rule-name>
```

The editable parameters are the same as those for creating export rules.

To view parameters of export rules created earlier, use the following command:

```
Admin@nodename# show logs logs-export
Admin@nodename# show logs logs-export <log-export-rule-name>
```

To remove previously created export rules, use the following command:

```
Admin@nodename# delete logs logs-export <log-export-rule-name>
```

To configure the parameters for a one-time log export, use the command:

```
Admin@nodename# execute logs send-once <log-export-rule-name>
<parameters>
```

Parameter	Description
fresh	Export fresh logs.
range	Specify the export range: • **start-export-range**: the start of the range in the format: 2022-12-31T23:59:59 • **end-export-range**: end of the range in the format: 2022-12-31T23:59:59

USERGATE APPLICATION AND SECURITY LANGUAGE (UASL)

General Information

UserGate Application and Security Language (UASL) is a language for writing user signatures and applications.

Custom signatures and applications can be added to IDPS profiles and application profiles to use in firewall rules.

Signature has the following structure:

```
UASL (.parameter1=<value1>; .parameter2=<value2>; …)
```

Signature parameters are specified in parentheses using semicolon (;) as a delimiter.

Multiline input can also be used:

```
UASL (.parameter1=<value1>;
      .parameter2=<value2>;
      ...
      )
```

> **Note**
> **Maximum length of custom signature is 1024 bytes.**

All conditions of one signature without exceptions will be combined using **AND** logical operator.

Created, edited and removed signatures can be tracked using event log record details.

Metainformation

The **.rev** field contains additional information: user who created signature, and signature creation date, type and version. This is optional field which does not affect IDPS activity; it can be used to track changes.

The field has the following format:

```
.rev = <date>,<version>,<status>,<author>;
```

and parameters have the following data types:

- <date>: integer;
- <version>: integer;
- <status>: string;
- <author>: string.

ID

The .id field contains signature or application ID.

```
.id=<id_value>
```

This field is optional. ID can also be set in signature or application details.

> **Note**
> **The value set in signature details on the *General* tab has a priority over the value set using UASL.**

Signature ID can range from 1000000 to 1049999; application ID can range from 1050000 to 1099999.

If the ID is set manually, its value is not unique and might be repeated.

If the ID was not set by the administrator, UserGate will assign it automatically (and when it is assigned automatically, its values might not be repeated). When the ID pool is exhausted, an error is displayed.

Filtering by IP Address

These parameters allow to configure checking of specified IP addresses:

```
.src_addr[!]=<IP_address/subnet>;

.dst_addr[!]=<IP_address/subnet>;
```

The following IP address formats can be used:

- A.B.C.D
- A.B.C.D/E
- A.B.C.D:E.

When specifying an IP address, it is not necessary to specify a network mask; to specify multiple IP addresses, use parentheses, for example:

```
.src_addr=[<IP_address1>, <IP_address2>];
```

Filtering by Port

To set TCP/UDP port checking, use the following parameters:

- **.src_port**: to check source ports
- **.dst_port**: to check destination ports

The following expressions are supported:

Name	Description
.src_port=<port_number>; .dst_port=<port_number>;	Specifying a certain source and/or destination port.
.src_port!=<port_number>; .dst_port!=<port_number>;	Checking all ports except for the specified one.
.src_port=:<port_number>; .dst_port=:<port_number>;	Checking all ports whose number is less than or equal to the specified port number.
.src_port!=:<port_number>; .dst_port!=:<port_number>;	Checking all ports whose number is greater than the specified port number.
.src_port=<port_number>:; .dst_port=<port_number>:;	Checking all ports whose number is greater than or equal to the specified port number.
.src_port!=<port_number>:; .dst_port!=<port_number>:;	Checking all ports whose number is less than the specified port number.
.src_port=<port_number1>:<port_number2>; .dst_port=<port_number1>:<port_number2>;	Checking all ports in the specified range (including port_number1 and port_number2).

Name	Description
.src_port!=<port_number1>:<port_number2>; .dst_port!=<port_number1>:<port_number2>;	Checking all ports which are not in the specified range (i. e. ports with numbers less than port_number1 and greater than port_number2).

Scanning Packets Without Payload

The **.nopayload** field allows to scan packets which do not have payload.

> **Note**
> **This field cannot be used for signatures with template lookup and tag checking.**

For example, this field can be used to find ports which have packets without payload. The signature for SYN scan detection is as follows:

```
UASL(.protocol=tcp; .tcp.flags=S; .rate=1000,2; .track=src_ip; .nopayload;)
```

> **Note**
> **Specifying the *.nopayload* field does not mean that the packets with payload are not scanned.**

Template Lookup

The following parameter allows to specify a template against which the IPS will scan the packet payload:

```
.pattern[!]="string";
```

> **Note**
> **Note that the search is case sensitive.**

HEX data should be specified using the "**|**" character. For example: |05 00 27|.

To specify special symbols, use the notations provided in the following table:

Symbol	HEX notation
"	|22|.
;	|3B| or |3b|.
\	|5C| or |5c|.
|	|7C| or |7c|.
:	|3A| or |3a|.

In addition to the **=** operator, the **!=** operator can also be used. If the latter operator is specified, it will search for packets which do not contain a specified template.

The parameter has the following general format:

```
.pattern[!]="string"; [.where=<MODE>;] [.no_case;] [.distance=<RANGE>[
,<MODE>];] [.within=<RANGE>[,<MODE>];] [.service=<MODE>;]
```

Search area modifiers (**.where**, **.no_case**, **.distance**, **.within**, **.service**) will be detailed later.

When writing a signature a number of **.pattern** parameters can be used to reduce the number of false positives.

Search Area Modifiers

.no_case

no_case modifier allows to perform case insensitive search in accordance with **.pattern** parameter.

.where

.where modifier is used to specify signature search area:

```
.where=<MODE>;
```

where <MODE> can take following values:

Name	Description
packet_origin	Search area is the whole packet without a protocol decoder.
uri	Search area is URI field of HTTP header.
host	Search area for HTTP session is the Host field (before line breaks).
header	The search area for an HTTP session is http headers, smtp and pop3 commands, tls and ssh protocol headers.
body	Search area is the body of HTTP packets.
file	The search area is decoded http content, eml attachments, ftp-data sessions.

.service

This search area modifies is needed to select a dissector.

```
.service=<MODE>;
```

where <MODE> can take following values:

Name	Description
http	HTTP protocol parsing.

.distance, .within, .at, .startin

These modifiers allow:

Name	Description
.distance	Skip specified number of bytes (RANGE) from the start or from the last found block. It has the following format: `.distance=<RANGE> [,<MODE>];` where <RANGE> — integer starting from 0. Optional parameters (<MODE>) will be detailed below. For example, the next record specifies skipping 10 bytes from the start for the first template and from the last found block for the second and subsequent templates. `.distance=10;` Example of using optional parameters: • skip 10 bytes from the start: `.distance=10, start;` • skip 10 bytes from the last found template: `.distance=10, match;`
.within	Scan selected interval (RANGE) from the start or from the last found block (the pattern falls completely within the specified range). It has the following format: `.within=<RANGE> [,<MODE>];` where <RANGE> — integer starting from 1. Optional parameters (<MODE>) will be detailed below. For example, the next record specifies the search from the 1th to 10th byte from the start for the first template and from the last found block for the second and subsequent templates. `.within=10;` Example of using optional parameters: • the search from the 1st (from the start) to the 10th byte:

Name	Description
	`.within=10, start;` • the search within 10 bytes from the last found template: `.within=10, match;`
.startin	Scan selected interval (RANGE) from the start or from the last found block (for a match, only the beginning of the pattern can fall within the specified range). It has the following format: `.startin=<RANGE> [,<MODE>];` where <RANGE> — integer starting from 1. Optional parameters (<MODE>) will be detailed below.
.at	Checking for presence of the template at the specified position. **Important!** This modifier cannot be used with **.distance** and **.within** modifiers. It has the following format: `.at=<RANGE> [,<MODE>];` where <RANGE> — integer starting from 0. Optional parameters (<MODE>) will be detailed below.

Optional parameters are provided in the table below:

Name	Description
start	Search from the beginning of data flow. **Important!** It is default value for the first template.
packet	Scan from the beginning of the packet.
reverse	Search from the end of the packet (useful for Next Protocol checking in ESP).
match	Search from the last found template. **Important!** It is default value for the second and subsequent templates.

Name	Description
lastmark	Scan from the last tag set using **.mark pset**.

> **Note**
> **If the optional parameter for .distance and .within modifiers has the same value, then the value of .within modifier is counted from the value of .distance modifier.**

For example, the record

```
.distance=10,match; .within=5,match
```

specifies the search in the range from 10th to 15th byte from the last found template.

.protocol

This modifier allows to specify a transport level protocol to which the signature will be applied:

```
.protocol=<MODE>;
```

where <MODE> can take following values:

- **icmp**: ICMP protocol traffic analysis
- **udp**: UDP protocol traffic analysis
- **tcp**: TCP protocol traffic analysis

> **Note**
> **Only one protocol can be specified. If no protocol is specified, then the signature will be applied only to TCP and UDP traffic.**

Triggering Frequency

If the frequency is set, the IDPS signature will be triggered not with every match, but only after a specified number of matches is detected for a specified time interval. This parameter can be useful, for example, to write signatures for detecting brute force attacks.

To specify the triggering frequency:

```
.rate=<count>, <period>;
```

here: <count> — number of triggered events

<period> — time interval (in seconds) during which the specified number of triggered events should occur.

The next parameter is optional and specifies the parameter for grouping matches:

```
.track=<MODE>;
```

here: <MODE> — the property which specifies the mode for packet tracking.

<MODE> can take following values:

- **src_ip**: tracking by source IP address
- **dst_ip**: tracking by destination IP address.

If **.track** modifier is not specified, then all matches are counted, and after the specified limit is reached, the signature is triggered.

Example:

```
UASL(.name="pop3.brute.force"; .protocol=tcp; .pattern="USER"; .flow=fr
om_server; .rate=3,60; .track=src_ip;)
```

The signature will be triggered after the USER template (.pattern="USER";) is discovered in packets sent from the same IP address (.track=src_ip;) more than 3 times for 60 seconds (.rate=3, 60;).

Analysis Direction

This parameter is optional and allows to apply the signature to certain traffic flows. It allows to create signatures that will analyze traffic from client, from server or in both directions.

```
.flow=<MODE>;
```

where <MODE> can take following values:

Name	Description
from_client	Analyze traffic from client.
from_server	Analyze traffic from server.
bi_directional	Analyze traffic in both directions.

Binary Data Search

The **.byte_test** parameter allows to compare a byte with a specified value and is applied to data presented in binary or character format.

The general format is as follows:

```
.byte_test = <bytes>,<operator>,<value>,<offset>,[,<multiplier>]
[,<modifiers>];
```

Available parameters are provided in the table:

Name	Description
<bytes>	Number of bytes in the current position with the specified offset which are read from the packet. Can take the following values: 1, 2 or 4.
<size>	String length; specified for string data.
*	Use all characters till the first non-numeric character.

Name	Description
<operator>	Operator used to compare the byte with the specified value: • **<** — less than • **>** — greater than • **=** — equal To; • **!=** — not equal to • **&** — the result of the logical "AND" operation between <bytes> and <MASK> (a number which specifies the bits of interest) is not equal to 0 • **~** — the result of the logical "AND" operation between <bytes> and <MASK> is equal to 0 • **^** — the result of the logical "XOR" operation between <bytes> and <MASK> is not equal to 0. Example: `.byte_test=1,&,0x80,0;` checks that the most significant bit of the first byte in the data field of the packet is set to 1.
<value>	The value used in comparison or packet size. The value can be specified using **0x** prefix; arithmetic operators (**+**, **-**, *****, **/**) can also be used.
<offset>	Offset in the data field of the packet: • **relative**: from the last match point. If the offset parameter is not specified, then the analysis, by default, is performed from the beginning of the packet.
<post_offset>	The number of bytes to move the scan start point. **Important!** Applied to **.byte_jump**.
<multiplier>	A numeric value by which the extracted number should be multiplied before comparison or moving the scan start point; this parameter is optional.
<modifiers>	Modifiers (optional): • **big** — process data from the most significant bit • **little** — process data from the least significant bit • **string** — the packet contains string data • **hex** — convert the data string to a hexadecimal number • **dec** — convert the data string to a decimal number

Name	Description
	• **oct** — convert the data string to a octal number • **align** — round the number of converted bytes to the next 32-bit boundary; used for **.byte_jump** only (for example, 0 ➜ 0; 1,2,3,4 ➜ 4; 5,6 ➜ 8 etc.).

Example: comparing the first four bytes of each packet with the value of 1234: packet data have character format in decimal numeration system:

```
.byte_test=4,=,1234,0,string,dec;
```

The **.byte_jump** parameter moves the scan start point to the specified number of bytes. The general format for data processing from the most or the least significant bit (i.e. for **big** and **little** modifiers) is as follows:

```
.byte_jump = <bytes>,<offset>,<post_offset>[,<multiplier>]
[,<modifiers>];
```

For string data (**string** modifier):

```
.byte_jump = (<size> | *),<offset>,<post_offset>[,<multiplier>]
[,<modifiers>];
```

Working with Tags

Named tag can be specified for each data flow. It is specified in the following way:

```
.mark <parameter>=<value>;
```

here: <value> — tag name (in quotes "")

<parameter> can take the values provided in the table below.

Pattern matching in most cases is bases on working with data packets. Tags are used when an attack pattern exists in a number of packets. The signature triggered for the

previous packet can add a tag; tags are checked when sending packets within one session.

Name	Description
set	Set the named tag for the current data flow.
pset	Set and remember the last added tag, so that it could be used with **.distance** and **.within** search area modifiers.
clear	Remove the named tag.
toggle	Change the status of the tag.
test	Check if the tag exists.
reset	Reset all tags.

Protocol Analyzers

In this section the fields of ICMP, TCP, UDP and HTTP protocols will be described.

Since only one protocol can be specified for one signature, using the following parameters will cause automatic protocol determination, i. e. it is the same as using **.protocol** parameter.

> **Note**
> **Specifying parameters from different protocols will cause an error.**

ICMP

The following parameters can be used to check ICMP header properties:

Name	Description
.icmp.type	Checking ICMP type. The following operators are supported: **=**, **!=**.
.icmp.code	Checking ICMP code value. The following operators are supported: **=**, **!=**.

Name	Description
.icmp.id	Checking ICMP ID value. The following operators are supported: **=**, **!=**.
.icmp.checksum	Verifying the checksum which is used when errors are detected. The following operators are supported: **<**, **>**, **<=**, **>=**, **=**, **!=**.
.icmp.data_size	Checking the size of the data field of the packet. This parameter is used to detect packets of abnormal size which are often used to cause buffer overflow. The following operators are supported: **<**, **>**, **<=**, **>=**, **=**, **!=**. When multiple conditions are set, they are combined using **AND** logic al operator.

TCP

The following parameters can be used to check TCP header properties:

Name	Description
.tcp.sport	Checking source port number or port range. The following operators are supported: **=**, **!=**.
.tcp.dport	Checking destination port number or port range. The following operators are supported: **=**, **!=**.
.tcp.window_size	Checking TCP window size. The following operators are supported: **<**, **>**, **<=**, **>=**, =, **!=**.
.tcp.checksum	Verifying the checksum which is used to check for errors when sending and/or receiving a packet. The following operators are supported: **<**, **>**, **<=**, **>=**, **=**, **!=**.
.tcp.seq	Checking values of TCP sequential numbers. The following operators are supported: **<**, **>**, **<=**, **>=**, =, **!=**. The **relative** modifier can be used to check against the starting number of the sequence. Application: `.tcp.seq=<value>,relative;` here: <value> is the TCP sequential number.

Name	Description
.tcp.flags	Checking TCP flags: `.tcp.flags=[<mod>]<tcp_flags>;` here: <mod> — modifier <tcp_flags> — TCP flag which can be specified in character or numeric (hexadecimal or decimal) format. Flags: • **0**: flags are not set • **F**, **1**, **0X001**: FIN • **S**, **2**, **0X002**: SYN • **R**, **4**, **0X004**: RST • **P**, **8**, **0X008**: PSH • **A**, **16**, **0X010**: ACK • **U**, **32**, **0X020**: URG • **E**, **64**, **0X040**: ECE • **C**, **128**, **0X080**: CWR • **N**, **256**, **0X100**: NS Modifiers: • ***** — at least one of specified flags should be set, the rest of the flags are not checked • **+** — all specified flags should be set, the rest of the flags are not checked • **!** — all specified flags should be reset, the rest of the flags are not checked • **!0** — at least one (any) flag should be set **Important!**If no modifier is set, then all specified flags should be set (strict match), and the rest of the flags should be reset.
.tcp.data_size	TCP packet payload size (without headers). The following operators are supported: **<**, **>**, **<=**, **>=**, =, **!=**. It is possible to specify it as .data_size (in this case the parameter will be applied to TCP and UDP protocols).

UDP

The following parameters can be used to check UDP header properties:

Name	Description
.udp.sport	Checking source port number or port range. The following operators are supported: **=**, **!=**.
.udp.dport	Checking destination port number or port range. The following operators are supported: **=**, **!=**.
.udp.checksum	Verifying the checksum. The following operators are supported: **<**, **>**, **<=**, **>=**, **=**, **!=**.
.udp.data_size	UDP packet payload size (without headers). The following operators are supported: **<**, **>**, **<=**, **>=**, **=**, **!=**. It is possible to specify it as .data_size (in this case the parameter will be applied to TCP and UDP protocols).

HTTP

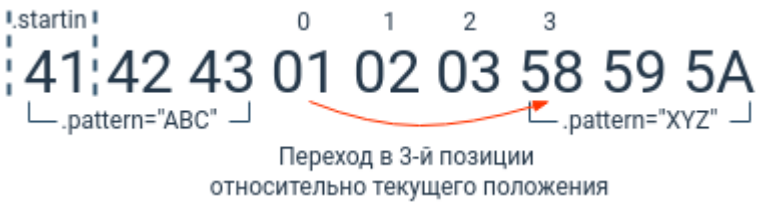
The following parameters can be used to check HTTP header properties:

Name	Description
.uri	Checking resource ID (URI) field.
.body	Checking the body of HTTP request or response.
.host	Checking node domain name.

Examples

In this sections a number of UASL code examples are provided.

Example 1



```
UASL(
    .id = 1;
    .pattern = "ABC"; .startin = 1;
    .pattern = "XYZ"; .at = 3, match;
)
```

In this example a sequential search for two patterns is performed:

- Search for the beginning of the ABC template within the specified range, i. e. the first byte of the specified template must fall within the range. The value of .startin modifier is 1 (range = 1, the template will be searched for from the beginning of the session, because the first modifier is "start" by default) — the byte falling within this range should be equal to the beginning of the ABC template.
- Search for XYZ template starting from position 4 (**.at=3;**) from the last found ABC template.

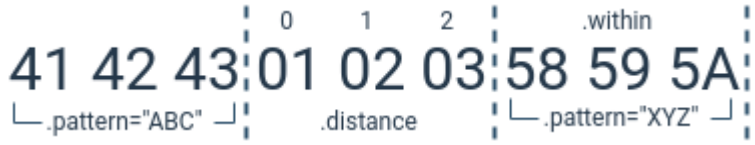
Example 2


```
UASL(
    .id = 2;
    .pattern = "ABC"; .at = 0;
    .pattern = "XYZ"; .distance = 3, match; .startin = 1, match;
)
```

A sequential search for two patterns is performed:

- Search for the beginning of the ABC template from the beginning of the packet (**.at=0;**).
- After skipping 3bytes from the last found ABC template (**.distance=3,match;**), search for the beginning of the XYZ template is performed; for the pattern to be triggered, the beginning of the XYZ template should match the first byte from which the search starts, because **.startin=1,match;**.

Example 3




```
UASL(
    .id = 3;
    .pattern = "ABC";
    .pattern = "XYZ"; .distance = 3, match; .within = 3, match;
)
```

A sequential search for two patterns is performed:

- Search for the ABC template;
- At 3 bytes after the last found ABC template (**.distance=3,match;**) the search for the XYZ template is performed: the XYZ template should completely fall within the following 3 bytes, because **.within=3**.

USERGATE POLICY LANGUAGE (UPL)

General Information

UPL stands for UserGate Policy Language. It is a language to describe UserGate policies (rule configurations used to make decisions about authentication requirements, access permissions, or content conversion).

General provisions of the UPL

You configure the rules using actions, conditions, and properties.

You specify one of the actions for each rule. Actions are settings that control transaction processing (OK, WARNING, PASS, DENY, FORCE_PASS, FORCE_DENY). When configuring rules that do not specify an action (for example, DNS, NAT and routing, traffic shaping, etc.), you must specify a PASS or OK action.

To specify conditions, use the equals (=) or not equal (!=) signs. You can use zones, addresses, source and destination GeoIP, services, applications, etc. All conditions in a rule are checked by logical AND, which means the rule works if all conditions are met.

Rule properties are specified in parentheses and are used to provide additional information, such as rule name, description, logging function, etc.

> **Note**
> **When configuring a rule, first specify the action, then the conditions, and then the properties.**

The UPL is used to create network and security policy rules for the following sections in the CLI:

- DNS proxy settings (**network dns dns-proxy dns-rules** level).
- Captive portal (**users captive-portal** level).
- Firewall (**network-policy firewall** level).
- NAT and routing (**network-policy nat-routing** level).
- Bandwidth (**network-policy traffic-shaping** level).
- Content filtering (**security-policy content-filtering** level).
- Safe browsing (**security-policy safe-browsing** level).
- Tunnel inspection (**security-policy tunnel-inspection** level).
- SSL inspection (**security-policy ssl-inspection** level).
- SSH inspection (**security-policy ssh-inspection** level).
- IDPS (**security-policy intrusion-prevention** level).
- Mail security (**security-policy mail-security** level).
- ICAP rules (**security-policy icap-rules** level).
- DoS rules (**security-policy dos-rules** level).
- Web portal (**global-portal web-portal** level).
- Reverse proxy rules (**global-portal reverse-proxy-rules** level).

- VPN server rules (**vpn server-rules** level).
- VPN client rules (**vpn client-rules** level).

To create a rule, use the following command:

```
Admin@nodename# create <level> <position> upl-rule <str-upl-syntax>
```

where <level> is the level at which the rule will be created,

<position> is the position at which the rule is located, and

<str-upl-syntax> is a string that describes the rule in UPL syntax.

To update an existing rule, use the following command:

```
Admin@nodename# set <level> <position> upl-rule <str-upl-syntax>
```

where <level> is the level at which the rule will be updated,

<position> is the number of the rule to update, and

<str-upl-syntax> is a string that describes the rule in UPL syntax.

To delete a rule, use the following command:

```
Admin@nodename# delete <level> <position | all>
```

where <level> is the level at which the rule will be deleted,

<position> is the number of the rule to delete, and

<all> means delete all rules.

To display a rule, use the following command:

```
Admin@nodename# show <level> <position | all>
```

<level> is the section for which to display the rules,

<position> is the number of the rule to display, and

<all> means display all rules.

Example of creating a firewall rule using UPL (multi-line input used):

```
Admin@nodename# create network-policy firewall 1 upl-rule \
...DENY \
...src.zone = Trusted \
...dst.zone = Untrusted \
...user = known \
...service = HTTPS \
...time = lib.time("Working hours") \
...rule_log(session)\
...name("Example of firewall rule created in CLI") \
...enabled(true)
```

Once the rule is created, it is displayed at the beginning of the firewall rules list (position 1). This rule denies HTTPS traffic from the Trusted zone to the Untrusted zone for users identified by the system. It works according to the "Working hours" schedule. When the rule triggers, the system logs information about the beginning of the session.

Comments

Any line starting with "**%**" is a comment.
A percent symbol "**%**" after a space or tab defines a comment that continues to the end of the line (unless the percent symbol appears inside quotation marks (""), then it is a part of an expression).

Example:

```
% This is a comment
DENY("Too many Host headers") request.header.Host.count = 2..  % and
this is a comment too
```

Comments can be placed anywhere in the policy description file.

Rules

The policy rule consists of conditions and a number of actions specified in any order. There are also properties that syntactically look like an action, but do not

perform active actions. For example, the *name* property simply adds a "name" attribute on the rule.

Rules are usually written on one line, but can be broken into lines using the special backslash character "****".

When a rule is executed, the condition is checked for the current specific transaction. If the condition evaluates to *True*, all listed actions are performed and the current layer ends with the prefixes *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*. If the triggered rule does not have the prefixes *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*, then actions are performed and then the next rule is processed. If the condition evaluates to *False* for this transaction, then the next rule is processed further.

All conditions in the rule are checked using logical "*And*". In other words, the rule will be triggered when all conditions are met.

In turn, a condition is a logical combination of triggers. The triggers are individual tests that can be executed with the components of a request and response, with the related users, or with the system state.

Actions are settings that control how a transaction is processed. For example, deny or process an object (rewrite the header: *rewrite*).

Syntax:

Rule ::= (PASS | FORCE_PASS | DENY| (DENY '(' string ')') | FORCE_DENY | FORCE_DENY'(' string ') '| WARNING | OK)? Conditions '\'? Actions

Conditions ::= condition '\'? Conditions

Actions ::= action '\'? Actions

Example:

The request will be denied when both triggers are activated:

- The domain will be example.com
- The time will be between 9 am and 5 pm

```
DENY url.domain = "example.com" time=09:00..17:00
```

Layers

A layer is a UPL construct used to group rules and make a single decision. Separate decision making helps control policy complexity. For that each decision is specified at a different layer.

Any rule at the layer can have the *PASS / FORCE_PASS / DENY / FORCE_DENY / OK / WARNING* prefix. When a rule with such prefix is triggered, all the rest rules at this layer are ignored.

If the rule with the *FORCE_PASS* or *FORCE_DENY* prefix is triggered, this means the processing is finished. Otherwise the next layer is processed. After all the layers have been processed, the request will be denied or passed depending on the last prefix — *PASS / FORCE_PASS* or *DENY / FORCE_DENY*. If the processing is stopped with the *WARNING* prefix, the warning will be added to the response body.

The *OK* prefix means that rule processing should be stopped at the current layer, if the corresponding conditions and actions (if specified) are met. If the prefix is missing when conditions and actions are executed, then no stop is implied.

The *FORCE_PASS* and *FORCE_DENY* actions are similar to the *PASS* and *DENY* actions, except that they can be overridden at subsequent layers. *FORCE_DENY* and *FORCE_PASS* immediately stop rule checking at both the current and subsequent layers, and this result is final.

Syntax:

Layer ::= '[' layer_type layer_name ']'

layer_type ::= ssl | ssh| captive| content | shaper | firewall | safebrowsing | dns | icap | mailsecurity | dos | webportal | reverseproxy | nat_routing | byod | vpn_server | vpn_client|idps | tunnel | scenarios | ipvs_server | icap_balancing | reverseproxy_balancing

layer_name ::= string

atom ::= [a-z][0-9a-zA-Z_]+

string ::= '"' 'arbitrary string '""

Example 1:

```
[content "L1"]
DENY enabled(true) % everything is disabled by default
```

```
[content "Devs"]
DENY group != Developers enabled(true)
%... next are the rules that will only apply to the Developers group
```

Example 2:

```
[content "Admin"]
FORCE_PASS group = Admins enabled(true)

[content "L2"]
DENY enabled(true) % everything is disabled by default
```

Dynamic Values

The values of "request address" (*url, url.host, url.path*), "source/destination IP address" (*src.ip, dst.ip*), "username" (*user*), "header values" (*request* and *response*), and "request parameters" (*qparam*) can be compared with each other and also used as arguments in actions where provided.

General Provisions

Comments

Any string starting with the "**%**" character is a comment.
If you use the "**%**" character after a space or tab symbol, the rest of the line becomes a comment (except for the cases when the percent character is inside the quotes ("") as part of an expression).

Example:

```
% This is a comment
DENY("Too many Host headers") request.header.Host.count = 2..  % and
this is a comment too
```

Comments can be placed anywhere in the policy description file.

Rules

A policy rule consists of conditions and a number of actions, written in any order. There are also properties that syntactically look like an action, but do not perform active actions. For example, the *name* property simply adds a "name" attribute on the rule.

Rules are usually written on one line, but can be broken into lines using the special backslash character "****".

When a rule is executed, the condition is checked for the current specific transaction. If the condition evaluates to *True*, all listed actions are performed and the current layer ends with the prefixes *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*. If the triggered rule does not have the prefixes *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*, then actions are performed and then the next rule is processed. If the condition evaluates to *False* for this transaction, then the next rule is processed further.

All conditions in the rule are checked using logical "*And*". In other words, the rule will be triggered when all conditions are met.

In turn, a condition is a logical combination of triggers. Triggers are individual tests that can be run against components of the request, response, associated users, or system state.

Actions are settings that control how a transaction is processed. For example, deny or process an object (rewrite the header: *rewrite*).

Syntax:

Rule ::= (PASS | FORCE_PASS | DENY| (DENY '(' string ')') | FORCE_DENY | FORCE_DENY'(' string ') '| WARNING | OK)? Conditions '\'? Actions

Conditions ::= condition '\'? Conditions

Actions ::= action '\'? Actions

Example:

The request will be denied when both triggers are fired:

- The domain will be example.com
- The time will be between 9 am and 5 pm

```
DENY url.domain = "example.com" time=09:00..17:00
```

Layers

A layer is a UPL construct used to group rules and make a single decision. Separate decision making helps control policy complexity. This is done by writing each solution in a separate layer.

Any rule in a layer can have the prefix *PASS / FORCE_PASS / DENY / FORCE_DENY / OK / WARNING*. When a rule with such a prefix is triggered, all other rules in the layer are skipped.

If a rule with the *FORCE_PASS* or *FORCE_DENY* prefix is triggered, then this is the final result of processing, otherwise processing moves to the next layer. After all layers have been processed, the request will be blocked or skipped depending on which was last: *PASS / FORCE_PASS* or *DENY / FORCE_DENY*. If processing stops at *WARNING*, a warning will be added to the response body.

The prefix *OK* implies stopping the processing of rules in the current layer when the conditions and actions (if specified) are met. If the prefix is missing when conditions and actions are executed, then no stop is implied.

The *FORCE_PASS* and *FORCE_DENY* actions are similar to *PASS* and *DENY*, except that they can be overridden at later layers. *FORCE_DENY* and *FORCE_PASS* immediately stop checking rules at both the current and subsequent layers, and this result is final.

Syntax:

Layer ::= '[' layer_type layer_name ']'

layer_type ::= ssl | ssh| captive| content | shaper | firewall | safebrowsing | dns | icap | mailsecurity | dos | webportal | reverseproxy | nat_routing | byod | vpn_server | vpn_client|idps | tunnel | scenarios | ipvs_server | icap_balancing | reverseproxy_balancing

layer_name ::= string

atom ::= [a-z][0-9a-zA-Z_]+

string ::= '"' arbitrary string '"'

Example 1:

```
[content "L1"]
DENY enabled(true) % everything is disabled by default

[content "Devs"]
DENY group != Developers enabled(true)
%... next are the rules that will only apply to the Developers group
```

Example 2:

```
[content "Admin"]
FORCE_PASS group = Admins enabled(true)

[content "L2"]
DENY enabled(true) % everything is disabled by default
```

Dynamic Values

The values of "request address" (*url, url.host, url.path*), "source/destination IP address" (*src.ip, dst.ip*), "username" (*user*), "header values" (*request* and *response*), and "request parameters" (*qparam*) can be compared with each other and also used as arguments in actions where provided.

Conditions

A condition in UPL is a logical combination of triggers. Triggers are individual tests that can be executed against request components, response components, associated users, or system state. All condition triggers are compared to values using the "=" and "!=" operators. The value can be a constant value, such as a string, an integer, a range of values, or a dynamic value.

Syntax:

condition ::= condition_name ('=' | '!=') condition_value

condition_value ::= pattern | list

list ::= '(' ((pattern ',') pattern)? ')'*

pattern ::= word | string | integer | float | boolean| range | condition_name

string ::= "" arbitrary string ""

*word ::= [a-zA-Z][0-9a-zA-Z_\-]**

boolean ::= yes|no|true|false

range ::= integer .. [integer] | [integer] .. integer | float .. [float] | [float] .. float

numeric :: = integer | range

http.connect

Checking for *HTTP CONNECT*.

Syntax:

http.connect = yes | no | true | false

http.method

Checking the HTTP method used. The method can be specified either in quotation marks or without.

Syntax:

http.method = GET | CONNECT | DELETE | HEAD | POST | PUT | TRACE | OPTIONS | TUNNEL | LINK | UNLINK | PATCH | PROPFIND | PROPPATCH | MKCOL | COPY | MOVE | LOCK | UNLOCK | MKDIR | INDEX | RMDIR | COPY | MOVE

http.request.version

Checking the HTTP request version.

Syntax:

http.request.version = 0.9 | 1.0 | 1.1

http.response.version

Checking the HTTP response version.

Syntax:

http.response.version = 0.9 | 1.0 | 1.1

http.response.code

Checking the HTTP response code. Valid values: 100 - 999.

Syntax:

http.response.code = NNN % (where NNN is a number from 100 to 999)

http.request.body, http.request.body.nocase, http.response.body and http.response.body.nocase

Checks the HTTP request/response body for a specific signature.

Example:

```
DENY http.response.body.nocase = "<title>index of" http.response.body =
">"
```

category

Checking whether a domain belongs to a certain category of sites (see the list of categories appendix).
A category can be specified both by its name and its identifier. If a category name contains spaces or special characters, it must be enclosed in quotation marks.

Syntax:

category = word | string | integer | list | lib

lib ::= lib.category '(' list_libs ')'

list_libs :: = lib_name ',' list_libs

lib_name ::= word | string

Example:

Disable the *Job Search* and *Gambling* categories:

```
DENY category = ("Job Search", Gambling)
```

Disable all categories from the *Restricted cats* library:

```
DENY category = lib.category("Restricted cats")
```

morphology

Checking the response body with morphological dictionaries.

Syntax:

morphology = word | string | list | lib

lib ::= lib.morphology '(' list_libs ')'

list_libs :: = lib_name ',' list_libs

lib_name ::= word | string

Example:

Disallow content if the morphology category from the *BadWords* dictionary is triggered:

```
DENY morphology = BadWords
```

Disable morphological categories from the **Special Words**, **BadWords** libraries:

```
DENY morphology = ("Special Words", BadWords)
DENY morphology = lib.morphology("Special Words", BadWords) % similar
to the previous rule
```

request.header.<h_name> and response.header.<h_name>

Checking HTTP request/response header. *h_name* can have one of the supported values (for the list of the supported HTTP headers refer to the Appendix).

Syntax:

request.header.<h_name>[.base64][.nocase] = string

Example:

```
DENY url="http://usergate.com" request.header.Pragma="no-cache"

PASS request.header.User-Agent = lib.useragent("Browsers")
PASS request.header.Content-Type = lib.mime("Applications")
DENY request.header.Connection.substring = "Upgrade"
```

request.header.<h_name>.substring and response.header.<h_name>.substring

Checking HTTP request/response header for a substring. *h_name* can have one of the supported values (for the list of the supported HTTP headers refer to the Appendix).

Syntax:

request.header.<h_name>[.base64]substring[.nocase] = string

Example:

```
DENY request.header.User-Agent.substring = "curl/"
```

request.header.<h_name>.regex and response.header.<h_name>.regex

Validate HTTP request/response header against PCRE regular expression. *h_name* can have one of the supported values (for the list of the supported HTTP headers refer to the Appendix).

Syntax:

request.header.<h_name>[.base64].regex = string

Example:

```
DENY("Accept only digits in content length") request.header.Content-
Length.regex != "[0-9]*"
```

request.header.<h_name>.re2 and response.header.<h_name>.re2

Validate HTTP request/response header against RE2 regular expression. *h_name* can have one of the supported values (for the list of the supported HTTP headers refer to the Appendix).

Syntax:

request.header.<h_name>[.base64].re2 = string

Example:

```
DENY("Accept only digits in content length") request.header.Content-
Length.re3 != "[0-9]*"
```

request.header.<h_name>.count and response.header.<h_name>.count

Checking the number of <*h_name*> headers in an HTTP request/response. *h_name* can have one of the supported values (for the list of the supported HTTP headers refer to the Appendix).

Syntax:

request.header.<h_name>.count = integer | range

Example:

```
DENY("Too many Host headers") request.header.Host.count = 2..
```

request.header.<h_name>.length and response.header.<h_name>.length

Check the length of all header values <*h_name*> in the HTTP request/response. *h_name* can have one of the supported values (for the list of the supported HTTP headers refer to the Appendix).

Syntax:

request.header.<h_name>.length = integer | range

Example:

```
DENY("Too much Cookie data") request.header.Cookie.length = 2048..
```

request.header_names, request.header_values, response.header_values and response.header_values

Check the name/value of all HTTP request/response headers for the value.

Syntax:

request.header_values[.base64].regex = string

request.header_values[.base64].re2 = string

request.header_values[.base64].substring[.nocase] = string

request.header_values.count = integer | range

request.header_values.length = integer | range

request.x_header.<xh_name> and response.x_header.<xh_name>

Check HTTP request/response header for value. *xh_name* is an arbitrary HTTP header.

Syntax:

request.x_header.<xh_name>[.base64][.nocase] = string

request.x_header.<xh_name>[.base64].regex = string

request.x_header.<xh_name>[.base64].re2 = string

request.x_header.<xh_name>[.base64].substring[.nocase] = string

request.x_header.<xh_name>.count = integer | range

request.x_header.<xh_name>.length = integer | range

Example:

```
DENY url="http://usergate.com" request.x_header.Test="test1"
```

The suffixes **length**, **count**, **regex**, **re2** are also possible, as in the case of <*h_name*>.

```
DENY("Too much X-Test data") request.x_header.X-Test.length = 2048..
DENY("Too much X-Test2 headers data") request.x_header.X-Test2.count =
2..
PASS request.x_header.Test.regex = "[0-9]*"
```

request.header.Cookie.<cookie_name>

Checking the Cookie request header for value.

Syntax:

request.header.Cookie.<cookie_name>[.base64][.(nocase | substring | substring.nocase | regex | re2)] = string

Example:

```
DENY http.method = POST request.header.Cookie.csrf_token !=
qparam.CSRF_TOKEN enabled(true) name("Check CSRF")
```

time, day, hour, minute

Checking whether the current time meets a given condition. If the suffix *utc* is not specified, local time is taken, otherwise — Greenwich Mean Time.

Syntax:

day[.utc] = monday | tuesday | wednesday | thursday | friday | saturday | sunday | DD | list

time[.utc] = HH:MM | range | lib.time(<name>)

hour[.utc] = HH | range

minute[.utc] = MM | range

HH ::= 00 - 23

MM ::= 00 - 59

DD ::= 1 - 31

Example:

```
PASS time = 12:00..13:00 % allow every day from 12:00 to 1:00 PM
PASS time = lib.time("Holidays") % use the "Holidays" library
DENY day = (sunday, saturday) % disable on weekends
DENY day = (monday, 15) hour = 9..18 % disable every Monday and every
15th of the month from 9:00 AM to 6:00 PM
```

Open intervals are counted according to the day/hour boundary.

```

PASS hour = 18.. % means that it is allowed from 6 PM to midnight
minute = ..10 % for the first 10 minutes of each hour
```

url, url.host and url.address

Checking a URL or part of it for value. The check uses a normalized URI with *%* decoded.

Syntax:

url[.(prefix | substring | suffix | regex | re2)] = string

url.host[.(prefix | substring | suffix | regex | re2)] = string

url.domain[.(prefix | substring | suffix | regex | re2)] = string

url.address = ip_address | subnet | subnet_label

url.port = [low_port]..[high_port] | port

url.path[.base64][.(prefix | substring | suffix | regex | re2)] = string

url.is_absolute = yes | no % whether or not the URL is absolute

prefix ::= string % the beginning of the string

substring ::= string % substring

suffix ::= string % the end of the string

regex ::= string % PCRE regular expression

re2 ::= string % RE2 regular expression

url.address: this is, in fact, a synonym for *dst.ip*.

Example:

```
DENY url.path.base64.re2 = "(?i)\bondisconnecting\W*=" enabled(true)
name("ondisconnecting (URI)")
```

qparam.<name>, qparam.values and qparam.names

Checking the value of request parameters. The check uses parameter names and values with decoded *%*.

Syntax:

qparam.length = numeric % check the total length of query parameters

qparam.count = numeric % check the number of query parameters
qparam.<name>[.(length | count)] = numeric

qparam.<name>[.base64][.(nocase | substring | substring.nocase | regex | re2)] = string

qparam.values[.base64].substring[.nocase] = string % check all values for substring

qparam.names[.base64].substring[.nocase] = string % check all names for substring

qparam.values[.base64].regex = string % check all values for regular expression

qparam.names[.base64].regex = string % check all names for regular expression

qparam.values[.base64].re2 = string % check all values for regular expression

qparam.names[.base64].re2 = string % check all names for regular expressions numeric ::= integer | range % number or range

regex ::= string % PCRE regular expression

re2 ::= string % RE2 regular expression

Example:

```
DENY("limit arguments total length") qparam.length =
1024..                % total
DENY("Limit argument value length") qparam.values.length =
1024..          % for each
DENY("Limit argument name length") qparam.names.length =
1024..            % for each
DENY("Maximum number of arguments in request limited") qparam.count  =
12.. % total
DENY(""PHP injection attempt") qparam.values.base64.substring.nocase =
"${@print"
```

user and group

Check the current user or his group.

Syntax:

user = word | string | known | unknown

group = word | string

user.guid = string

group.guid = string

known: used to indicate an authorized (known) user;
unknown: used to indicate an unauthorized (unknown) user.

Example:

```
PASS user = known % allow known users
DENY group = "Sales Department" category = "Pornography/Sexually
Explicit" % deny the porn category for the "Sales Department" group
```

src and dst

Checking the condition for the source/destination IP address, zone or GeoIP.

Syntax:

src.ip = ip_address | subnet | subnet_label | list | lib

dst.ip = ip_address | subnet | subnet_label | list | lib

src.zone = integer | zone_name

dst.zone = integer | zone_name

src.geoip = iso3166 | list

dst.geoip = iso3166 | list

src.mac = mac_address | list

dst.mac = mac_address | list

lib ::= lib.(network | url) '(' list_libs ')'

list_libs :: = lib_name ',' list_libs

lib_name ::= word | string

iso3166 ::= [A-Z][A-Z]

url.address: this is, in fact, a synonym for *dst.ip*.

scenario

Checking if a specific scenario is active.

Syntax:

scenario = string | word | list

Example:

```
DENY scenario = Torrents  desc("Deny if the Torrents script is active")
```

virus_heuristic and virus_usergate

Checking the response body for viruses.

Syntax:

virus_heuristic = yes | no | true | false % (false by default)

virus_usergate = yes | no | true | false % (false by default)

heuristic: heuristic analyzer (slow);
usergate: hash checker (fast).

bridge_vlan_filter

Filtering traffic by VLAN tags for an interface in bridge mode.

Syntax:

bridge_vlan_filter '='|'!=' list | number | number..number

Example:

```
DENY bridge_vlan_filter = (10, 100..200)  desc("Deny traffic by VLAN
tags")
```

service

Detecting traffic of a specific service for a firewall.

Syntax:

service = string | word | list

Example:

```
DENY service = POP3  desc("Disable POP3 service")
```

application

Detecting traffic of a specific application at L7 level for firewall.

Syntax:

application = string | word | list

Example:

```
DENY application = Tor  desc("Deny Tor")
```

envelope_from and envelope_to

Checking the email address of the sender/recipient of the message.

Syntax:

envelope_from '='|'!=' string | list

envelope_to '='|'!=' string | list

Example:

```
PASS envelope_from = "Email froup from" envelope_to = "Email froup to"
service = SMTP mark_hdr(Subject) enabled(true) name("Mail Pass Rule")
```

response_time

Checking response time in milliseconds.

Syntax:

response_time = integer

hip_profile

HIP profiles for checking endpoint compliance. Valid for firewall rules only.

Syntax:

hip_profile = string | word | list

Built-in Libraries

Libraries (*lib*) are UPL language elements that are used to access built-in and user libraries. These lists are usually rather large, which makes describing them using **def** definitions ineffective. Libraries are accessed by their names.

Syntax:

library ::= lib.<url | morphology | category | useragent | mime | network | time | applicationgroup | servicegroup>(list_names)

list_names ::= name list_names

name ::= word | string

url: URL list;
morphology: list of morphological dictionaries;
category: category group;
useragent: list of useragents;
mime: list of content types;
network: list of networks/IP addresses;
time: library with time intervals.

Example:

```
DENY src.ip = lib.network("Bad ips", "Test ips")
DENY dst.ip = lib.network("Bad ips")
DENY dst.ip = lib.url("Bad urls") % in this case, domains will be
resolved to IP addresses

DENY morphology = lib.morphology("Porno words", "Bad words")
DENY category = lib.category("Restricted categories") category =
lib.category(Productivity)

PASS request.header.User-Agent = lib.useragent("Browsers")
PASS request.header.Content-Type = lib.mime(Applications)
DENY time = lib.time(Weekends)
```

Definitions

In the policy files the definitions (**def**) are used to combine the sets of conditions or actions. Each definition must have a unique username by which it can be referenced from rules.

def condition

Sets of conditions. All conditions in one line are checked by logical *AND*. Line feed means logical *OR*. The escape character is a backslash ("****") at the end of a line that allows you to move the condition by **AND** to the next line.

Syntax:

def condition label_name

conditions

end

conditions ::= condition '\'? [conditions]

condition ::= name '=' value

label_name ::= atom

atom ::= [a-z][0-9a-zA-Z_]+

def scenario_cond

The list of scenario conditions. Each condition of a scenario is usually written on one line, but if necessary, the condition can be split into lines using a special character, the backslash ("****").

Syntax:

def scenario_cond label_name

scenario_conditions

end

scenario_conditions ::= Conditions '\'? Properties

Conditions ::= condition '\'? Conditions

Properties ::= property '\'? Properties

scenario_cond ::= name '=' value

label_name ::= atom

atom ::= [a-z][0-9a-zA-Z_]+

def var

Definition of variables. Serves to count certain events over a certain period of time. To change the value, use the actions **inc** and **dec**.

Syntax:

```
def var label_name
   init ::= integer
   window ::= time
   key ::= condition_name | condition_list
end
label_name ::= atom
atom ::= [a-z][0-9a-zA-Z_]+
condition_list ::= '(' condition_name , condition_list ')'
```

init is the initial value of the variable, to which it will return after **window** time has passed;

key is the field or list of fields by which the variable values are grouped (optional).

Properties

Properties are certain attributes of a rule, such as *name* or *enabled*. They are used to provide additional information during rule processing. The syntax for properties is exactly the same as for actions.

Syntax:

```
property = prop_name | prop_name '(' list_params ')'
prop_name ::= name | desc | id | rule_log | enabled | scenario
list_params ::= value ',' list_params
```

name and desc

The *name* and *description* attributes for the rule.

Syntax:

```
Name ::= name '(' string|word ')'
```

Description ::= desc '(' string ')'

Example:

```
DENY hour = 9..18 category = News name("Deny News")  desc("Deny
category News during working hours")
```

enabled

An attribute that enables or disables the rule.

Syntax:

Enable ::= enabled '(' boolean ')'

boolean ::= yes | no | true | false % (false by default)

rule_log

Sets the logging attribute of a rule. The *session* value is only valid for firewall, DOS protection, and bandwidth rules.

Syntax:

Logging ::= rule_log '(' boolean | session ')'

LoggingFwRule ::= rule_log '(' boolean , interval, burst')'

boolean ::= yes | no | true | false % (no by default)

interval ::= "integer/[s,m,h,d]"

burst ::= integer

interval: average number of packets matching the *limit* condition per unit of time (1/s, 1/m, 1/h, 1/d), default = 3/h;
burst: maximum number of packets matching the *limit* condition at one time (default = 5).

profile

Sets the rule profile.

Syntax:

Profile ::= profile '(' string | word | list ')'

certificate

The certificate used to support HTTPS connections. Valid for reverse proxy rules only.

Syntax:

CertAuthEnabled ::= cert_auth_enabled '(' boolean ')'

Certificate ::= certificate '(' certificate_name ')'

certificate_name ::= string | word

gateway

Gateway. Name of one of the existing gateways. Valid only for NAT and routing rules and for Health Check scenario conditions.

Syntax:

Gateway ::= gateway '(' string | word ')'

Firewall Rule Properties

reject_with

Sets the method by which traffic will be blocked. Valid for firewall rules only.

Syntax:

Reject ::= reject_with '(' "tcp-reset-both" | "tcp-rst" | "host-unreach" ')'

fragmented

Checking for packet fragmentation. Valid for firewall rules only.

Syntax:

Fragmented ::= fragmented '(' boolean ')'

boolean ::= yes | no | true | false

yes: only fragmented packets are checked;
no: only unfragmented packets are checked;
—: if the *fragmented* property is not specified, all packets will be checked.

ips_profile

Sets the IPS profile. Valid for firewall rules only.

Syntax:

IPS_Profile ::= ips_profile '(' string | word ')'

l7_profile

Sets the application profile. Valid for firewall rules only.

Syntax:

L7_Profile ::= l7_profile '(' string | word ')'

SSL inspection rule properties

block_invalid_cert

Blocks sites with invalid certificates. Valid for SSL inspection rules only.

Syntax:

InvalidCertificate ::= block_invalid_cert '(' boolean ')'

boolean ::= yes | no | true | false

check_revoc_cert

Checks certificate revocation list. Valid for SSL inspection rules only.

Syntax:

ChekRevocation ::= check_revoc_cert '(' boolean ')'

boolean ::= yes | no | true | false

block_expired_cert

Blocks expired certificates. Valid for SSL inspection rules only.

Syntax:

ExpiredCertificate ::= block_expired_cert '(' boolean ')'

boolean ::= yes | no | true | false

block_self_signed_cert

Blocks self signed certificates. Valid for SSL inspection rules only.

Syntax:

SelfSignedCertificate ::= block_self_signed_cert '(' boolean ')'

boolean ::= yes | no | true | false

ssl_profile

SSL profile. Valid only for SSL inspection rules, reverse proxy, web portal.

Syntax:

SslProfile ::= ssl_profile '(' string | word ')'

ssl_forward_profile

SSL forwarding profile. Valid for SSL inspection rules only.

Syntax:

SslForwardProfile ::= ssl_forward_profile '(' string | word ')'

profile_id ::= integer

Traffic Shaping Rule Properties

bandwidth_pool

Bandwidth pools. Valid only for traffic shaping rules.

Syntax:

BandwidthPool ::= bandwidth_pool '(' bandwidth ')'

bandwidth ::= "100 Kbps" | "512 kbps" | "1 Mbps" | "2 Mbps" | "5 Mbps" | "10 Mbps" | "20 Mbps" | "50 Mbps" | "100 Mbps"

Example:

```
scenario = "Torrent Detection Scenario" bandwidth_pool("1 Mbps")
enable(true) name("Torrent Bandwidth")
```

Properties of Mail Traffic Protection Rules

mark

Mark. The text of the tag used to mark emails. Valid for mail traffic protection rules only.

Syntax:

Mark ::= mark '(' string | word ')'

mark_hdr

Header. The field where the marking tag is placed. Valid for mail traffic protection rules only.

Syntax:

MarkHeader ::= mark_hdr '(' word ')'

antispam_kav

Antispam check. Valid for mail traffic protection rules only.

Syntax:

AntispamKav ::= antispam_kav '(' boolean ')'

boolean ::= yes | no | true | false

antispam_usergate

UserGate antispam check. Valid for mail traffic protection rules only.

Syntax:

AntispamUsergate ::= antispam_usergate '(' boolean ')'

boolean ::= yes | no | true | false

Example:

```
DENY("with error") envelop_to = UserGate antispam_usergate(yes)
```

dnsbl

DNSBL check (SMTP only). Valid for mail traffic protection rules only.

Syntax:

DNSBLacklistCheck ::= dnsbl '(' boolean ')'

boolean ::= yes | no | true | false

Example:

```
DENY service = SMTP dnsbl(yes)
```

Properties of NAT and Routing Rules

target_ip

DNAT destination address if *dnat* or *port_mapping* action is set.

New IP network/mask if action *netmap* is specified.

Valid for NAT and routing rules only.

Syntax:

TargetIp ::= target_ip '(' ipv4 | ipv4_with_mask ')'

Example:

```
PASS target_ip("192.168.1.20") dnat
```

target_snat

Enabling SNAT. If enabled, UserGate will replace the source address in the packets from the external network with its own IP address.

Valid for NAT and routing rules only.

Syntax:

TargetSnat ::= target_snat '(' boolean ')'

boolean ::= yes | no | true | false

snat_target_ip

SNAT IP address (external IP). Explicitly sets the IP address with which the source address will be replaced when replacing packet addresses.

Valid for NAT and routing rules only.

Syntax:

SnatTargetIp ::= snat_target_ip '(' ip_address ')'

port_map

Port forwarding. Port overrides for published services. Valid for NAT and routing rules only.

Syntax:

PortMap ::= port_map '(' protocol, port_from, port_to ')'

protocol ::= tcp | udp | smtp | smpts

port_from ::= integer

port_to ::= integer

port_from: TCP/UDP port number to which users send requests;
port_to: TCP/UDP port number to which user requests to the internal published server will be forwarded.

direction

Direction of network substitution. Valid for NAT and routing rules only.

Syntax:

Direction ::= direction '(' input | ouput ')'

input: input, replace the destination IP network address. Destination IP addresses in the traffic will be substituted.
output: output, replace source IP network address. Source IP addresses in the traffic will be substituted.

DNS Proxy Rules Properties

dns_server

List of DNS server IP addresses. Valid for DNS proxy rules only.

Syntax:

DnsServer ::= dns_server '(' ip_address | ip_address_list ')'

Reverse Proxy Rules Properties

cert_auth_enabled

Authentication via certificate. Valid for reverse proxy rules only.

Syntax:

CertAuthEnabled ::= cert_auth_enabled '(' boolean ')'

boolean ::= yes | no | true | false

is_https

Enables HTTPS support. Valid for reverse proxy rules only.

Syntax:

IsHttps ::= is_https '(' boolean ')'

boolean ::= yes | no | true | false

rewrite_path

Path rewrite. Valid for reverse proxy rules only.

Syntax:

RewritePath ::= rewrite_path '(' path_from, path_to ')'

path_from: change from (the URL domain and/or path that needs to be substituted);
path_to: change to (the URL domain and/or path with which the original ones should be substituted).

waf_profile

WAF profile. Valid for reverse proxy rules only.

Syntax:

WafProfile ::= waf_profile '(' string | word | list ')'

Web Portal Rules Properties

icon

Icon to display on the web portal for this bookmark. Valid for web portal rules only.

Syntax:

Icon ::= icon '(' string | word ')'

additional_url

Supporting URLs necessary for the main URL to work (but not needed to be published to users).

Valid for web portal rules only.

Syntax:

AdditionalUrl ::= additional_url '(' string | word | list ')'

rdp_check_session_alive

Checking authorization for RDP sessions. Valid for web portal rules only.

Syntax:

RdpCheckSessionAlive ::= rdp_check_session_alive '(' boolean ')'

boolean ::= yes | no | true | false

transparent_auth

Enables transparent user authentication. Valid for web portal rules only.

Syntax:

TansparentAuth ::= transparent_auth '(' boolean ')'

boolean ::= yes | no | true | false

Web Security Rules Properties (Safe Browsing)

enable_adblock

Block advertising (AdBlock).

Syntax:

EnableAdblock ::= enable_adblock '(' boolean ')'

boolean ::= yes | no | true | false

url_list_exclusions

URL list of exception sites for which ad blocking is not required.

Syntax:

UrlListExclusions ::= url_list_exclusions '(' list_libs ')'

list_libs :: = lib_name ',' list_libs

lib_name ::= word | string

enable_injector

Enables you to insert the desired code into all web pages.

Syntax:

EnableInjector ::= enable_injector '(' boolean ')'

boolean ::= yes | no | true | false

custom_injector

Injector code.

Syntax:

CustomInjector ::= custom_injector '(' string ')'

safe_search

Safe search feature.

Syntax:

SafeSearch ::= safe_search '(' boolean ')'

boolean ::= yes | no | true | false

search_history_logging

Logging user search requests.

Syntax:

SearchHistoryLogging ::= search_history_logging '(' boolean ')'

boolean ::= yes | no | true | false

social_sites_block

Blocking apps in popular social networks.

Syntax:

SocialSitesBlock ::= social_sites_block '(' boolean ')'

boolean ::= yes | no | true | false

BYOD Properties

max_device_number

The maximum number of devices from which a user can access the network.
Only valid for BYOD rules.

Syntax:

MaxDeviceNumber ::= max_device_number '(' integer ')'

max_active_device_number

The maximum number of devices from which a user can simultaneously access the network.
Only valid for BYOD rules.

Syntax:

MaxActiveDeviceNumber ::= max_active_device_number '(' integer ')'

device_type

The type of devices for which this BYOD policy rule applies.
Only valid for BYOD rules.

Syntax:

DeviceType ::= device_type '(' word | string | list ')'

approving_required

Administrator confirmation.
Only valid for BYOD rules.

Syntax:

ApprovingRequired ::= approving_required '(' boolean ')'

boolean ::= yes | no | true | false

SSH Inspection Properties

block_ssh_shell

Blocks SSH remote shell. Valid for SSH inspection rules only.

Syntax:

BlockSshShell ::= block_ssh_shell '(' boolean ')'

boolean ::= yes | no | true | false

block_ssh_exec

Blocks SSH remote execution. Valid for SSH inspection rules only.

Syntax:

BlockSshExec ::= block_ssh_exec '(' boolean ')'

boolean ::= yes | no | true | false

block_sftp

Blocking SFTP (Secure File Transfer Protocol) connection. Valid for SSH inspection rules only.

Syntax:

BlockSftp ::= block_sftp '(' boolean ')'

boolean ::= yes | no | true | false

ssh_command

The linux command to pass, in the format ssh user@host 'command'.

Valid for SSH inspection rules only.

Syntax:

SshCommand ::= ssh_command '(' string ')'

VPN Server Rules Properties

auth_profile

Auth profile. Valid for VPN server rules only.

Syntax:

AuthProfile ::= auth_profile '(' string | word ')'

vpn_network

VPN network. Valid for VPN server rules only.

Syntax:

VpnNetwork ::= vpn_network '(' string ')'

interface

VPN Interface. Valid for VPN server and VPN client rules only.

Syntax:

Interface ::= interface '(' word ')'

VPN Client Rules Properties

server_address

The server's IP address. Valid for VPN client rules only.

Syntax:

ServerAddress ::= server_address '(' ip_address ')'

password

Password. Valid for VPN client rules only.

Syntax:

Password ::= password '(' word ')'

last_error

VPN last error. Information field not available for editing. Valid for VPN client rules only.

Syntax:

LastError ::= last_error '(' string ')'

connection_time

Connection time. Information field not available for editing. Valid for VPN client rules only.

Syntax:

ConnectionTime ::= connection_time '(' word ')'

status

Status. Information field not available for editing. Valid for VPN client rules only.

Syntax:

Status ::= status '(' string ')'

Scenario Rule Properties

trigger

Type of scenario triggering. Valid for scenario rules only.

Syntax:

Trigger ::= trigger '(' trigger_type ')'

trigger_type ::= all_users | one_user

duration

Duration. The time in minutes for which the scenario will remain activated. Valid for scenario rules only.

Syntax:

Duration ::= duration '(' integer ')'

operation_mode

The scenario is triggered when one or all conditions are met.

Syntax:

OperationMode ::= operation_mode '(' mode ')'

mode ::= all | any

Scenario Condition Properties

scond_type

Type of scenario conditions. Valid for scenario conditions only.

Syntax:

SCondType ::= scond_type '(' type ')'

SCondType ::= scond_type '(' type ')'

count_interval

The number of triggered alerts. Valid only for the "URL category" (url_category), "Application" (app), "Content type" (mime_type), and "Health check" (health_check) scenario conditions.

Syntax:

CountInterval ::= count_interval '(' integer ')'

max_event_count

The time interval in minutes during which *count_interval* triggers were detected. Valid only for the "URL category" (url_category), "Application" (app), "Content type" (mime_type), and "Health check" (health_check) scenario conditions.

Syntax:

MaxEventCount ::= max_event_count '(' integer ')'

packet_size

Packet size. The packet size in the user's traffic has exceeded the set value Valid only for the "Packet size" (net_packet_size) scenario conditions.

Syntax:

PacketSize ::= packet_size '(' size ')'

size ::= integer | integer KB | integer MB | integer GB

64 — size in bytes;
2MB — size in megabytes.

traffic_limit

Traffic limit. Valid only for the "Traffic limit" (traffic) scenario conditions.

Syntax:

TrafficLimit ::= traffic_limit '(' size ')'

size ::= integer | integer KB | integer MB | integer GB

period

Period. Valid only for the "Traffic limit" (traffic) scenario conditions.

Syntax:

Period ::= period '(' time_period ')'

time_period ::= minute | hour | day | week | month

ips_tl

IDPS threat level. Valid only for "IDPS" scenario conditions (ips).

Syntax:

IpsTl ::= ips_tl '(' level ')'

level ::= integer | very_low | low | medium | high | very_high

ips_tl(2) scond_type(ips);
ips_tl(low) scond_type(ips).

health_check_method

Checking method. Valid only for the "Health check" (health_check) scenario conditions.

Syntax:

CheckMethod ::= health_check_method '(' method ')'

method ::= ping | dns | get

health_result

Result. Valid only for the "Health check" (health_check) scenario conditions.

Syntax:

esult ::= health_result '(' result ')'

result ::= positive | negative

health_request_timeout

Connection timeout (sec). Valid only for the "Health check" (health_check) scenario conditions.

Syntax:

RequestTimeout ::= health_request_timeout '(' integer ')'

health_answer_timeout

Response timeout (sec). Valid only for the "Health check" (health_check), "HTTP GET verification method" (get) scenario conditions.

Syntax:

ut ::= health_answer_timeout '(' integer ')'

health_type_request

DNS query type. Valid only for the "Health check" (health_check), "DNS verification method" (dns) scenario conditions.

Syntax:

TypeRequest ::= health_type_request '(' type_name ')'

type_name ::= word | string

Load Balancing Rule Properties

scheduler

Scheduler. Valid for TCP/UDP load balancing rules only.

Syntax:

Scheduler ::= scheduler '(' balancing_type ')'

balancing_type ::= wrr | rr | lc | wlc

wrr — Weighted round robin;

rr — Round robin;

lc — Least connections;

wlc — Weighted least connections.

real_server

Real servers. Valid for TCP/UDP load balancing rules only.

Syntax:

RealServer ::= real_server '(' mode, ipv4[:port], weight ')'

mode ::= gate | masq | masq_snat

port ::= integer

weight ::= integer

gate: gateway;

masq: masquerading;

masq_snat: masquerading with source IP address substitution (SNAT).

Example:

```
OK \
        url.address = 172.168.13.100 \
        url.port = 10000 \
        service = tcp \
        scheduler(wlc) \
        real_server(masq_snat, 172.168.13.11:10000, 50) \
        ipvs_fallback(gate, 172.168.13.12)
        monitor_kind(ping) \
        monitor_interval(60) \
        monitor_timeout(60) \
        monitor_failurecount(10) \
        enabled(true) \
        name("TCP/UPD load balancing")
```

ipvs_fallback

Server fallback mode. Valid for TCP/UDP load balancing rules only.

Syntax:

IpvsFallback ::= ipvs_fallback '(' mode, ipv4[:port] ')'

mode ::= gate | masq | masq_snat

port ::= integer

monitor_kind

Mode: Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorKind ::= monitor_kind '(' kind ')'

kind ::= ping | connect | negotiate

monitor_service

Service. Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorService ::= monitor_service '(' service ')'

service ::= http | dns

monitor_request

Request. Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorRequest ::= monitor_request '(' string ')'

monitor_response

Expected response. Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorResponse ::= monitor_response '(' string ')'

monitor_interval

Check interval. Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorInterval ::= monitor_interval '(' integer ')'

monitor_timeout

Check timeout. Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorTimeout ::= monitor_timeout '(' integer ')'

monitor_failurecount

Max failures. Valid for TCP/UDP load balancing rules only.

Syntax:

MonitorFailurecount ::= monitor_failurecount '(' integer ')'

Actions

An action is what will be performed if the conditions in a rule are true. Parameters can be constant values, or dynamic values where provided.

Syntax:

action = action_name | action_name '(' list_params ')'

action_name ::= warning | log_message | append | delete | set | replace | encrypt | inc| dec | reset | redirect | encrypt_body_url | decrypt_path | body_inject | set_cookie_token | body_replace | lookup_and_auth | encode_cookie | decode_cookie | sma | nat | dnat | route | port_mapping | netmap | forward | ignore | action_label

action_label ::= 'action'.<action_label_name>

action_label_name ::= atom

list_params ::= value ',' list_params

warning

Actions that are allowed without exiting rule processing at the current layer. *warning* marks the need to insert warning code into the response body.

Example:

```
[L1]
category = lib.category(Productivity) warning
DENY user = user1
```

log_message

Log messages.

Example:

```
DENY category = lib.category(Productivity) log_message("Deny porno")
```

append

Add a header to the HTTP request/response. For the list of the supported headers refer to the Appendix.

If a header is included into one *request* or *response* group, the first parameter can be omitted.

Syntax:

append([request | response,] <headername>, value)

headername: see the Appendix.
value ::= string | numeric | condition_name.

set

Overwrite the value of a specific HTTP header. For the list of the supported headers, refer to the Appendix.
If a header is included into one *request* or *response* group, the first parameter can be omitted.

Syntax:

set([request | response,] <headername>, value)

headername: see the Appendix.
value ::= string | numeric | condition_name.

delete

Remove HTTP header. For the list of the supported headers refer to the Appendix.

If a header is included into one *request* or *response* group, the first parameter can be omitted.

Syntax:

delete([request | response,] <headername>)

headername: see the Appendix.

replace

Modify the value of an HTTP header. For the list of the supported headers refer to the Appendix.

If a header is included into one *request* or *response* group, the first parameter can be omitted.

Syntax:

replace([request | response,] <headername>, regex, value)

regex ::= string % regular expression

value ::= string | condition_name

headername: see the Appendix.

Example 1:

Add the Referer header:

```
PASS append(Referer, "http://example.com") enabled(true)
```

Remove the header:

```
PASS delete(Referer)
```

Rewrite the header:

```
PASS set(request, Cache-Control, no-cache)
```

Modify the Location header:

```
PASS response.header.Location.count = 1.. replace(response, Location,
"http://example.com", url.host) enabled(true)
```

Example 2:

```
define action delete_referer
    log_message("Referer header deleted")
    delete(request, Referer)
end
```

encrypt

Encrypt the path portion of the HTTP header. For the list of the supported headers, refer to the Appendix.
If a header is included into one *request* or *response* group, the first parameter can be omitted.
The encryption key and the "Use the IP address as part of the encryption key" flag are optional.

Syntax:

encrypt([request | response,] <headername>, <url>[, <user_key>[, <add_ip>]])

url ::= string % part of url for filtering

user_key ::= string % user encryption key (optional)

add_ip ::= boolean % whether to add IP address to the encryption key (boolean, optional)

boolean ::= yes | no | true | false

headername: see the Appendix.

encrypt_body_url

Encrypt the part of the path in the response body references.
The encryption key and the "Use the IP address as part of the encryption key" flag are optional.

Syntax:

encrypt_body_url(<url>[, <user_key>[, <add_ip>]])

url ::= string % part of url for filtering

user_key ::= string % user encryption key (optional)

add_ip ::= boolean % whether to add IP address to the encryption key (boolean, optional)

boolean ::= yes | no | true | false

decrypt_path

Decrypt part of the request path.

If a header is included into one *request* or *response* group, the first parameter can be omitted.

The encryption key and the "Use the IP address as a part of the encryption key" flag are optional.

Syntax:

decrypt_path(<path>[, <user_key>[, <add_ip>]])

path ::= string % part of the path for filtering

user_key ::= string % user encryption key (optional)

add_ip ::= boolean % whether to add IP address to the encryption key (boolean, optional)

boolean ::= yes | no | true | false

Example:

Encrypt all relative paths in the *Location* header and response body, and decrypt the request path:

```
decrypt_path("/", "User_Key", true) enabled(true) name("Path decode")
http.response.code = 302 encrypt(Location, "/", "User_Key", true)
enabled(true) name("Encrypt Location header")
encrypt_body_url("/", "User_Key", true) enabled(true) name("Encrypt all
relative URL")
```

body_inject

Insert the script into the response body.

Syntax:

body_inject(inject_text)

inject_text ::= string

set_cookie_token

Add the 'Set-Cookie' header with the generated token to the response.

Syntax:

set_cookie_token(cookie_name, parameter, expires_date)

cookie_name ::= string

parameter ::= string

expires_date ::= [DD_]HH:MM % time that will be added to the current time

Example:

Implementation of CSRF protection:

```
DENY http.method = POST request.header.Referer.substring = "/login.php"
qparam.UCSRF_TOKEN != request.header.Cookie.ucsrf_token enabled(true)
name("Check CSRF")
url.path.prefix = "/login.php" set_cookie_token(ucsrf_token, "path=/",
01_00:00) body_inject("<script language='JavaScript'>
    var tokenName = 'UCSRF_TOKEN';

    document.addEventListener('DOMContentLoaded', function()
    {
        var t_res = document.cookie.match(/ucsrf_token=(.+?)(;|$)/);
        var tokenValue = t_res ? t_res[1] : '';

        var forms = document.getElementsByTagName('form');
        for(i=0; i<forms.length; i++)
        {
```

```
            var html = forms[i].innerHTML;
            html += '<input type=hidden name=' + tokenName + ' value='
+ tokenValue + ' />';
            forms[i].innerHTML = html;
        }
    });
</script>") enabled(true) name("Inject")
```

encode_cookie

Encrypt cookie values in the *Set-Cookie* header with the given name.

Syntax:

encode_cookie(cookie_name[, condition_name][, user_kry_string][, f_encrypt])

cookie_name ::= string

condition_name % the condition used for encoding (default src.ip)

user_kry_string ::= string % user encryption key (default "")

f_encrypt := true % encryption required (false by default)

decode_cookie

Decrypt the token in the Cookie header with the given name.

Syntax:

decode_cookie(cookie_name[, condition_key][, user_kry_string][, f_decript])

cookie_name ::= string

condition_name % the condition used for decoding (default src.ip)

user_kry_string ::= string % user encryption key (default "")

f_encrypt := true % encryption required (false by default)

Example:

Encrypting and decrypting a cookie named *security:*

```
response.header.Set-Cookie.count != 0 encode_cookie("security", src.ip,
true) enabled(true) name("encode_cookie")
request.header.Cookie.count != 0 decode_cookie("security", src.ip,
true) enabled(true) name("decode_cookie")
```

body_replace

Modify the response body. No more than two (first) modifications are performed for each answer.

Syntax:

body_replace(<regex>, <value>)

regex ::= string % regular expression

value ::= string

Example:

```
PASS \
body_replace("(\\+7|8)[\\s(]?(\\d\{3})[\\s)]?(\\d\{3})[\\s-]?(\\d\{2})
[\\s-]?(\\d\{2})", "+\\1 (\\2) \\3-XX-XX") \
body_replace("(\\w{1})[\\w\.]*(\\w{1})@([\\w]+)\\.([\\w]+)", "\\1***\
\2@\\3.\\4") \
enabled(true) \
name("Replace mail and phone")
```

lookup_and_auth

Authenticate a user. If the IP is not specified, the request is marked with the username.

Syntax:

lookup_and_auth(<user_login>[, <ip_address>[, <session_timeout>]])

user_login ::= string | condition_name % Authentication login

ip_address ::= string | condition_name % IP address

session_timeout ::= integer % session timeout by default 0.

Example:

```
lookup_and_auth(request.x_header.X-Authenticated-User,
request.x_header.X-Forwarded-For, 300) enabled(true) name("User
authentication")
lookup_and_auth(request.x_header.X-Authenticated-User) enabled(true)
name("Mark request")
```

redirect

When blocked, redirect the user to the address specified in the redirect.

Syntax:

Redirect ::= redirect(RespCode[, RedirectText], Url)

RespCode ::= 301 | 302 | 305 | 307

RedirectText ::= string

Url ::= string

Example:

```
DENY src.zone = Trusted redirect(302, "Custom test (Moved)", "https://
block.captive/block")
DENY src.zone = Untrusted redirect(302, "https://block.captive/block")
```

inc and dec

Used to change the value of variables declared as *def var*.

Syntax:

inc(var.<var_name>, integer)

dec(var.<var_name>, integer)

Example:

For every *http.response.code = 500* the *rps* value increases by 1. If we exceed 10 such requests within 5 minutes, we block further responses. After 5 minutes, the *rps* variable will be reset to 0:

```
def var rps
    init = 0
    window = 00:05
    key = src.ip
end

http.response.code = 500 var.rps=..10 inc(var.rps, 1) enabled(true)

PASS var.rps = 5 log_message("Warning!") enabled(true)

DENY var.rps=11.. log_message("Too many 500 errors!") enabled(true)
```

reset

Reset the values of variables declared as *init* in *def var* to their initial values.

Syntax:

reset(var.<var_name>)

sma

Used to calculate the average value over a time window, which is defined in the variable as *window* in *def var*.

Syntax:

sma(var.<var_name>, integer)

Example:

Requests are blocked when the average request time over a 30-second interval exceeds 2 seconds:

```
def var avg_time
    init = 0
    window = 00:00:30
```

```
    key = src.ip
end

src.zone = Untrusted sma(var.avg_time, response_time) enabled(true)
name("sma")
DENY src.zone = Untrusted var.avg_time = 2000.. enabled(true) name("sma
res")
```

nat

NAT is a substitution of network IP addresses.

Example:

```
PASS src.zone = Trusted dst.zone = Untrusted nat
```

dnat

DNAT is a substitution of the destination IP address.

Example:

```
PASS dnat target_ip("192.168.1.20")
```

port_mapping

Port forwarding is the redirection of traffic to a specified IP address by changing the port number of the published service.

Example:

```
PASS port_mapping target_ip("192.168.1.20") port_map(tcp, 2000, 2001)
```

netmap

Network mapping allows to replace source or destination IPs from one network to another.

Example:

```
PASS netmap target_ip("192.168.32.0/24") direction(output)
```

route

Policy-based routing: allows to route IP packets based on extended information, for example, services, MAC addresses, or servers (IP addresses).
SNAT IP: an address which will be used as source IP for NAT traffic.

Example:

```
PASS service = HTTP route getway(Gateway1)
```

ignore

Ignore the response from the ICAP server. In this case, regardless of the ICAP server's response, the data is sent to the user unmodified.

Example:

```
OK profile("Example ICAP server") enabled(true) name("ICAP rule")
ignore
```

forward

Forward. If the SSL/TLS traffic is successfully decrypted, a copy of the traffic will be forwarded in accordance with the SSL inspection rule and profile.

Example:

```
OK url = lib.url(ZAPRET_INFO_BLACK_LIST_DOMAIN)ssl_profile("Default SSL
profile") ssl_forward_profile("Forward SSL profile") enabled(false)
name("Decrypt rule") forward
```

Rule Types

Content filtering rules

Prefixes

Name	Description
PASS	Permission to visit a web page.
DENY	Blocking a web page.
WARNING	Warning the user that visiting this page is undesirable.

Conditions

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

category, scenario, time, url, user.

request.header.Referer: list of URLs that specify referrers for the current page.
request.header.User-Agent: Useragent of user browsers.
response.header.Content-Type: lists of content types.
http.method: method used in HTTP requests.

Properties

name, desc, enabled, rule_log, virus_heuristic.

Example

```
[content "Content Rules"]
 % ----------------- 1 --- "Content Rules" --------------
PASS \
    url = lib.url("Education institutions") \
    src.zone = Trusted \
    desc("Content filtering rule which allows access to the list of
URLs. This is an example rule which can be changed or deleted if
necessary.") \
    rule_log(yes) \
```

```
    enabled(true) \
    name("Example white list")
% ---------------- 2 --- "Content Rules" -------------
DENY("Blockpage (EN)") \
    url = lib.url("Education institutions") \
    dst.zone = Untrusted \
    rule_log(yes) \
    enabled(false) \
    name("Example block RU RKN by URL list")
% ---------------- 3 --- "Content Rules" -------------
DENY \
    url = lib.network("Private IPs") \
    morphology = lib.morphology(Drugs) \
    src.zone = Trusted \
    time = lib.time(Weekdays) \
    rule_log(yes) \
    redirect(302, "https://bing.com") \
    enabled(false) \
    name("Example redirect to safesearch engines")
```

Firewall Rules

Prefixes

Name	Description
PASS	Allowing traffic.
DENY	Blocking traffic.

Conditions

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

service, scenario, time, url, user, hip_profile.

Properties

name, desc, enabled, rule_log, reject_with, ips_profile, l7_profile.

Example

```
[firewall "Firewall rules"]
% ---------------- 1 ----------------
DENY \
    scenario = "Example torrent detection scenario" \
    dst.zone = Untrusted \
    dst.ip = lib.network("Botnets IP list") \
    rule_log(session) \
    reject_with("host-unreach") \
    enabled(true) \
    name("Example block RU RKN by IP list")
% ---------------- 2 ----------------
PASS \
    scenario = "Example torrent detection scenario" \
    src.zone = Trusted \
    dst.zone = Untrusted \
    hip_profile = "HIP profile" \
    ips_profile("Default IDPS profile") \
    l7_profile("Pass all applications") \
    rule_log(yes, "3/h", 5) \
    enabled(true) \
    name("Allow trusted to untrusted")
```

NAT and Routing Rules

Prefixes

Name	Description
OK	Always OK.

The rule type is determined by the action (Action):

- nat: NAT rule;
- dnat: DNAT rule;
- port_mapping: port forwarding rule;
- netmap: Network mapping rule;
- route : policy-based rule.

Conditions

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

service, scenario, time, url, user.

Properties

name, desc, enabled, rule_log, direction, target_ip, target_snat, snat_target_ip,

port_map, gateway.

Example

```
[nat_routing "NAT and Routing Rules"]
% ----------------- 1 -----------------
OK \
    src.zone = Trusted \
    service = (HTTP, HTTPS) \
    snat_target_ip("192.168.13.210") \
    rule_log(session) \
    enabled(true) \
    name("NAT Rule")\
    nat
% ----------------- 2 -----------------
OK \
    src.zone = Management \
    dst.ip = (lib.network("Private IPs"), lib.url("Microsoft Windows
Internet checker")) \
    target_ip("171.168.1.1") \
    target_snat(yes) \
    snat_target_ip("192.168.1.1") \
    enabled(true) \
    name("DNAY Rule")\
    dnat
% ----------------- 3 -----------------
OK \
    target_ip("172.168.1.1") \
    snat_target_ip("192.168.1.1") \
```

```
    port_map(tcp, 2000, 2000) \
    enabled(true) \
    name("Port-forwarding Rule")\
    port_mapping
% ---------------- 4 ----------------
OK \
    user = example \
    scenario = "Example torrent detection scenario" \
    gateway(My) \
    enabled(true) \
    name("Policy-base Rule")\
    route
% ---------------- 5 ----------------
OK \
    dst.geoip = (RW, SO) \
    target_ip("172.168.1.1") \
    direction(input) \
    enabled(true) \
    name("Network mapping Rule")\
    netmap
```

Captive Portal Rules

Prefixes

Name	Description
PASS	Do not use authentication.
OK	Use a Captive profile.

Conditions

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

time, url, category.

Properties

name, desc, enabled, rule_log, profile.

Example

```
[captive "Captive Rules"]
% ---------------- 1 ----------------
PASS \
    category = lib.category(Threats) \
    url = lib.url("Microsoft Windows Internet checker") \
    time = lib.time(Weekends) \
    rule_log(yes) \
    enabled(true) \
    name("Skip auth for Microsoft Internet checker")
% ---------------- 2 ----------------
OK \
    src.zone = Trusted \
    profile("Example Captive profile") \
    enabled(true) \
    name("Example Captive portal")
```

SSL Inspection Rules

Prefixes

Name	Description
PASS	Do not decrypt transmitted data.
OK	Decrypt transmitted data.

Forwarding is determined by the action (Action):

- **forward: if the SSL/TLS traffic is successfully decrypted, a copy of the traffic will be forwarded in accordance with the SSL inspection rule and profile (ssl_forward_profile).**

Conditions

src.zone, src.geoip, src.ip,

dst.geoip, dst.ip,

time, service, user, category.

Properties

name, desc, enabled, rule_log, ssl_profile, ssl_forward_profile.

block_invalid_cert, check_revoc_cert, block_expired_cert, block_self_signed_cert.

Example

```
[ssl "Decrypt Rules"]
% ---------------- 1 ----------------
PASS \
    category = (Finance, "Information Security") \
    rule_log(yes) \
    ssl_profile("Default SSL profile") \
    enabled(false) \
    name("Example DO NOT Decrypt rule for Finance and Security sites")
% ---------------- 2 ----------------
OK \
    category = lib.category(Threats) \
    rule_log(yes) \
    block_invalid_cert(yes) \
    check_revoc_cert(yes) \
    block_expired_cert(yes) \
    block_self_signed_cert(yes) \
    ssl_profile("Default SSL profile") \
    enabled(false) \
    name("Example decrypt rule for parental control")
% ---------------- 3 ----------------
OK \
    url = lib.url("Default SSL profile") \
    rule_log(yes) \
    ssl_profile("Default SSL profile") \
    ssl_forward_profile("SSL forward profile") \
    enabled(false) \
    name("Example decrypt RU RKN")\
    forward
```

SSH Inspection Rules

Prefixes

Name	Description
PASS	Do not decrypt transmitted data.
OK	Decrypt transmitted data.

Conditions

src.zone, src.geoip, src.ip,

dst.geoip, dst.ip,

time, service, user.

Properties

name, desc, enabled, rule_log,

block_ssh_shell, block_ssh_exec, block_sftp, ssh_command.

Example

```
[ssh "SSH inspection Rules"]
% ----------------- 1 -----------------
PASS \
    service = "Any UDP" \
    rule_log(yes) \
    enabled(true) \
    name("Bypass Rule")
% ----------------- 2 -----------------
OK \
    service = IMAP \
    block_ssh_shell(yes) \
    block_ssh_exec(yes) \
    block_sftp(yes) \
    ssh_command("command") \
    rule_log(yes) \
```

```
    enabled(true) \
    name("Decrypy Rule")
```

DNS Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

url.domain.

Properties

name, desc, enabled, dns_server.

Example

```
[dns "DNS Rules"]
% ----------------- 1 -----------------
OK \
    url.domain = "*.example.com" \
    dns_server(1.2.3.4) \
    enabled(true) \
    name("Dns rule")
```

DoS Rules

Prefixes

Name	Description
PASS	Allowing traffic.
DENY	Unconditional traffic blocking.
WARNING	Applying the DoS attack protection profile.

Conditions

src.zone, src.geoip, src.ip,

dst.zone, dst.geoip, dst.ip.

time, service, user, scenario,

Properties

name, desc, enabled, rule_log, profile.

Example

```
[dos "DoS Rules"]
% ---------------- 1 ----------------
PASS \
    scenario = "Example torrent detection scenario" \
    user = example \
    src.ip = lib.url("Microsoft Windows Internet checker") \
    dst.geoip = RW \
    service = FTP \
    rule_log(session) \
    enabled(true) \
    name("DoS Allow Rule")
% ---------------- 2 ----------------
DENY \
    desc(api_dos_rule) \
    rule_log(yes, "3/h", 5) \
    enabled(true) \
    name("DoS Deny Rule")
% ---------------- 3 ----------------
WARNING \
    user = "CN=VPN users,DC=LOCAL" \
    time = lib.time("Working hours") \
    profile("DoS Profile") \
    enabled(false) \
    name("DoS Protect Rule")
```

ICAP rules

Prefixes

Name	Description
PASS	Bypass: do not send the data to the ICAP server.
OK	Sending data to the ICAP server.

Action (Action):

- **ignore: ignore the response from the ICAP server.**

Conditions

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

url, category, user, service, http.method, response.header.Content-Type.

Properties

name, desc, enabled, rule_log, profile.

Example

```
[dos "DoS Rules"]
% ---------------- 1 ----------------
PASS \
    scenario = "Example torrent detection scenario" \
    user = example \
    src.ip = lib.url("Microsoft Windows Internet checker") \
    dst.geoip = RW \
    service = FTP \
    rule_log(session) \
    enabled(true) \
    name("DoS Allow Rule")
% ---------------- 2 ----------------
DENY \
    desc(api_dos_rule) \
```

```
    rule_log(yes, "3/h", 5) \
    enabled(true) \
    name("DoS Deny Rule")
% ---------------- 3 ----------------
WARNING \
    user = "CN=VPN users,DC=LOCAL" \
    time = lib.time("Working hours") \
    profile("DoS Profile") \
    enabled(false) \
    name("DoS Protect Rule")
```

Mail Traffic Protection Rules

Prefixes

Name	Description
PASS	Traffic passing without changes.
DENY ("with error")	Blocking the message and reporting an error in delivering the message to the server.
DENY	Blocking a message without notifying about the blocking.
WARNING	Marking of mail messages.

Conditions

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

user, service, envelope_from, envelope_to.

Properties

name, desc, enabled, rule_log, mark_hdr, mark, antispam_usergate, dnsbl.

Example

```
[mailsecurity "Mail Security Rules"]
% ---------------- 1 ----------------
```

```
PASS \
    user = (example, "CN=VPN users,DC=LOCAL") \
    envelope_from = "Email froup from" \
    envelope_to = "Email froup to" \
    service = SMTP \
    rule_log(yes) \
    mark_hdr(Subject) \
    enabled(true) \
    name("Mail Pass Rule")
% ----------------- 2 -----------------
DENY("with error") \
    service = (SMTPS, SMTP) \
    rule_log(yes) \
    mark_hdr(Subject) \
    antispam_usergate(yes) \
    enabled(true) \
    name("Mail Drop Rule")
% ----------------- 3 -----------------
DENY \
    src.zone = Untrusted \
    service = SMTP \
    mark_hdr(Subject) \
    dnsbl(yes) \
    enabled(false) \
    name("DNSBL spam drop rule")
% ----------------- 4 -----------------
WARNING \
    src.zone = Untrusted \
    service = (SMTP, POP3, SMTPS, POP3S) \
    mark_hdr(Subject) \
    mark("[SPAM]") \
    antispam_usergate(yes) \
    enabled(false) \
    name("SMTP and POP3 filtering")
```

Reverse Proxy Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

user, request.header.User-Agent, url.port.

Properties

name, desc, enabled, rule_log, profile,
certificate, cert_auth_enabled, is_https, ssl_profile,

waf_profile, rewrite_path.

Example

```
[reverseproxy "Reverse proxy Rules"]
% ---------------- 1 ----------------
OK \
    url.port = 80 \
    src.zone = Untrusted \
    desc("Example reverse proxy rule. This is an example rule which can
be changed or deleted if necessary. ") \
    profile("Example reverse proxy server") \
    rewrite_path("example.com/path1", "example.local/path1") \
    waf_profile("Example WAF profile") \
    enabled(true) \
    name("Example reverse proxy rule")
```

Web Security Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

time, user,

Properties

name, desc, enabled, rule_log.

enable_adblock, safe_search, search_history_logging, social_sites_block, enable_injector,

custom_injector, url_list_exclusions.

Example

```
[safebrowsing "Safe browsing Rules"]
% ----------------- 1 -----------------
OK \
    rule_log(yes) \
    enable_adblock(yes) \
    safe_search(yes) \
    search_history_logging(yes) \
    social_sites_block(yes) \
    enable_injector(yes) \
    custom_injector(code) \
    url_list_exclusions(FISHING_BLACK_LIST) \
    desc("Safebrowsing rule for all users. This is an example rule
which can be changed or deleted if necessary.") \
    enabled(false) \
    name("Example safebrowsing")
```

Traffic Shaping Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

time, user application, service, scenario,

Properties

name, desc, enabled, rule_log, bandwidth_pool.

Example

```
[shaper "Shaper Rules"]
% ----------------- 1 -----------------
OK \
    scenario = "Example torrent detection scenario" \
    service = "HTTP Proxy" \
    rule_log(session) \
    bandwidth_pool("1 Mbps") \
    enabled(true) \
    name("Example Bandwidth rule")
% ----------------- 2 -----------------
OK \
    scenario = "Example torrent detection scenario" \
    src.zone = Trusted \
    application = lib.category("Coin Miners", Business) \
    rule_log(yes, "3/h", 5) \
    bandwidth_pool("100 Kbps") \
    enabled(true) \
    name("Example torrent shaper")
```

Tunnel Inspection Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

service.

Properties

name, desc, enabled.

Example

```
[tunnel "Tunnel inspection Rules"]
% ----------------- 1 -----------------
PASS \
    src.zone = Trusted \
    dst.zone = Untrusted \
    service = gre \
    enabled(true) \
    name("Example Tonnel Incpection Bypass rule")
% ----------------- 2 -----------------
OK \
    dst.geoip = YE \
    service = gtpu \
    enabled(true) \
    name("Example Tonnel Incpection Rule")
```

Web Portal Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

url, user, url.domain.

Properties

name, desc, enabled.

icon, ssl_profile, certificate, additional_url, rdp_check_session_alive, transparent_auth.

Example

```
[webportal "Web portal Rules"]
% ----------------- 1 -----------------
OK \
    user = "CN=Default Group,DC=LOCAL" \
    url = "http://www.intranet.loc" \
    icon("default.svg") \
    rdp_check_session_alive(yes) \
    transparent_auth(yes) \
    certificate("CA (Default)") \
    ssl_profile("Default SSL profile") \
    enabled(false) \
    name("Example http application published via web portal")
```

VPN Server Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

src.zone, src.geoip, src.ip, src.mac.

dst.geoip, dst.ip.

user,

Properties

name, desc, enabled.

profile, auth_profile, vpn_network, interface.

Example

```
[vpn_server "VPN server Rules"]
% ---------------- 1 ----------------
OK \
    user = "CN=VPN users,DC=LOCAL" \
    src.zone = Untrusted \
    profile("Remote access VPN profile") \
    auth_profile("Example user auth profile") \
    vpn_network("Remote access VPN network") \
    interface(tunnel1) \
    enabled(false) \
    name("Remote access VPN rule")
```

VPN Client Rules

Prefixes

Name	Description
OK	Always OK.

Properties

name, desc, enabled.

profile, interface, server_address.

last_error, status, connection_time: information fields that cannot be edited.

Example

```
[vpn_client "VPN client Rules"]
% ---------------- 1 ----------------
OK \
    server_address("10.10.10.10") \
    last_error(Disabled) \
    status(disconnected) \
    connection_time(0) \
    profile("Client VPN profile") \
    interface(tunnel3) \
    enabled(true) \
    name("Client VPN rule")
```

Scenario Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

scenario_cond.

Properties

name, desc, enabled.

operation_mode, trigger, duration.

Scenario conditions

URL category (url_category)

Conditions

category

Properties

count_interval, max_event_count, scond_type

Virus detection (virus_detection)

Properties

scond_type

Application (app)

Conditions

application

Properties

count_interval, max_event_count, scond_type

IDPS (ips)

Properties

ips_tl, scond_type

Content type (mime_type)

Conditions

response.header.Content-Type

Properties

count_interval, max_event_count, scond_type

Packet size (net_packet_size)

Properties

packet_size, scond_type

Sessions per one IP (sessions_per_ip)

Properties

sessions_limit, scond_type

Traffic limit (traffic)

Properties

traffic_limit, period, scond_type

Health check (health_check)

Conditions

url.address, url.domain

Properties

health_check_method, health_result, health_request_timeout, health_type_request, health_answer_timeout, count_interval, max_event_count, scond_type

Example

```
[scenarios "Scenario Rules"]
% ---------------- 1 ----------------
def scenario_cond example_scenario_define
    category = (lib.category(Threats), "Advertisements & Pop-Ups")
count_interval(10) max_event_count(3) scond_type(url_category)
    scond_type(virus_detection)
    application = lib.category(Threats) count_interval(2)
max_event_count(1) scond_type(app)
    ips_tl(medium) scond_type(ips)
    response.header.Content-Type = lib.mime("Java script")
count_interval(0) max_event_count(0) scond_type(mime_type)
    packet_size(200MB) scond_type(net_packet_size)
    sessions_limit(50) scond_type(sessions_per_ip)
    traffic_limit(2GB) period(hour) scond_type(traffic)
    url.address = "192.168.100.100" url.domain = "example.com"
```

```
health_check_method(dns) \
        health_result(negative) health_request_timeout(4)
health_type_request(a) \
        count_interval(5) max_event_count(3) scond_type(health_check)
    url.domain = "example.com" health_check_method(get) \
        health_result(negative) health_request_timeout(5)
health_answer_timeout(10) \
        count_interval(0) max_event_count(0) scond_type(health_check)
end
OK \
    scenario_cond = example_scenario_define \
    operation_mode(all) \
    trigger(one_user) \
    duration(5) \
    enabled(false) \
    name("Example torrent detection scenario")
```

TCP/UDP Load Balancing Rules

Prefixes

Name	Description
OK	Always OK.

Conditions

src.zone, src.geoip, src.ip, src.mac.

service, url.address, url.port.

Properties

name, desc, enabled.

scheduler, real_server, ipvs_fallback

monitor_kind, monitor_service, monitor_request, monitor_response, monitor_interval,

monitor_timeout, monitor_failurecount

Example

```
[ipvs_server "TCP/UDP load balancing Rules"]
% ---------------- 1 ----------------
OK \
    src.geoip = RW \
    url.address = 192.168.1.100 \
    url.port = 80 \
    service = tcp \
    scheduler(rr) \
    real_server(gate, 1.1.1.1:80, 50) \
    ipvs_fallback(masq_snat, 8.8.8.8:10000) \
    monitor_kind(negotiate) \
    monitor_service(http) \
    monitor_request("example.com") \
    monitor_interval(60) \
    monitor_timeout(60) \
    monitor_failurecount(10) \
    enabled(true) \
    name("TCP load balancing")
```

ICAP Load Balancing Rules

Prefixes

Name	Description
OK	Always OK.

Properties

name, desc, enabled, profile.

Example

```
[icap_balancing "ICAP load balancing Rules"]
% ---------------- 1 ----------------
OK \
    profile("Example ICAP server") \
```

```
    enabled(true) \
    name("ICAP load balancing")
```

Reverse Proxy Load Balancing Rules

Prefixes

Name	Description
OK	Always OK.

Properties

name, desc, enabled, profile.

Example

```
[reverseproxy_balancing "Reverse proxy load balancing Rules"]
% ---------------- 1 ----------------
OK \
    profile("Example reverse proxy server") \
    enabled(true) \
    name("Reverse-proxy load balancing")
```

APPLICATIONS

List of Categories

ID	Name	Description	Description
0	Unknown	Not categorized	Not categorized

ID	Name	Description	Description
1	Advertisements and Pop-Ups	Sites that provide advertising graphics or other ad content files that appear on Web pages.	Sites that provide advertising graphics or other ad content files that appear on Web pages.
2	Alcohol and Tobacco	Sites that promote or sell alcohol- or tobacco-related products or services.	Sites that promote or sell alcohol- or tobacco-related products or services.
3	Anonymizers	Sites that act as an intermediary for surfing to other websites in an anonymous fashion, whether to circumvent web filtering or for other reasons.	Sites that act as an intermediary for surfing to other websites in an anonymous fashion, whether to circumvent web filtering or for other reasons.
4	Arts	Sites with artistic content or relating to artistic institutions such as theaters, museums, galleries, dance companies, photography, and digital graphic resources.	Sites with artistic content or relating to artistic institutions such as theaters, museums, galleries, dance companies, photography, and digital graphic resources.
5	Business	Sites that provide business related information such as corporate web sites. Information, services, or products that help businesses of all sizes to do their day-to-day commercial activities.	This category includes the corporate websites and the Internet resources containing information on the services and products, which are needed by the businesses of all sizes to operate successfully.
6	Transportation	Sites that include information about	Websites related to motor vehicles and

ID	Name	Description	Description
		motor vehicles such as cars, motorcycles, boats, trucks, RVs and the like, including online purchase sites. Includes manufacturer sites, dealerships, review sites, pricing, enthusiasts clubs, etc.	their purchase. This category includes the Internet resources of manufacturers, dealers, sites with reviews, communities, driving clubs etc.
7	Chat	Sites that enable web-based exchange of real-time messages through chat services or chat rooms	Websites, which allow for real-time communication.
8	Kids sites	Websites that are family and children oriented.	Websites for children.
9	Forums and Newsgroups	Sites for sharing information in the form of newsgroups, forums, bulletin boards. Does not include personal blogs.	News feeds, forums, and bulletin boards. This category does not include personal blogs.
10	Compromised	Sites that have been compromised by someone other than the site owner in order to install malicious programs without the user's knowledge. Includes sites that may be vulnerable to a particular highrisk attack.	Websites that have been taken over by cybercriminals to distribute malware without the owner's knowledge. The category also includes the resources with the high risk of infection.

ID	Name	Description	Description
11	Computers and Technology	Sites that contain information such as product reviews, discussions, and news about computers, software, hardware, peripheral and computers services.	Websites containing product reviews, news and information on computers, software, hardware, peripheral devices, technologies, and services.
12	Criminal Activity	Sites that offer advice on how to commit illegal or criminal activities, or to avoid detection. These can include how to commit murder, build bombs, pick locks, etc. Also includes sites with information about illegal manipulation of electronic devices, hacking, fraud and illegal distribution of software.	Websites providing tips and recommendations on committing illegal or criminal actions and containing information on how to hide the traces of crime. These websites contain instructions on how to commit a murder, make a bomb, pick locks etc. This category also includes websites related to the illegal use of electronic devices, hacking, fraud, and illegal software distribution.
13	Dating and Personals	Sites that promote networking for interpersonal relationships such as dating and marriage. Includes sites for match-making, online dating, spousal introduction, escort services.	Websites helping to establish personal relationships, such as dating and marriage. This category includes websites related to matchmaking, dating and escort services.

ID	Name	Description	Description
14	Download Sites	Sites that contain downloadable software, whether shareware, freeware, or for a charge. Includes some peer-to-peer sites.	Websites with software available for download. This category also includes some peer-to-peer websites.
15	Education	Sites sponsored by educational institutions and schools of all types including distance education. Includes general educational and reference materials such as dictionaries, encyclopedias, online courses, teaching aids and discussion guides	Websites supported by the educational institutions and schools of all types, including remote learning. It includes general educational and reference materials, such as dictionaries, encyclopedias, interactive courses, textbooks, and teaching materials.
16	Entertainment	Sites containing programming guides to television, movies, music and video (including video on demand), celebrity sites, and entertainment news.	Websites providing TV channels, movies, music, and video (including video on demand). This category also includes celebrity websites and entertainment news.
17	Finance	Sites related to banking, finance, payment or investment, including banks, brokerages, online stock trading, stock quotes, fund management, insurance companies, credit unions, credit card	Websites related to banks, finance, payments, or investments. This category includes websites of banks, trading firms, and online trading portals. These resources also provide information on

ID	Name	Description	Description
		companies, and so on.	stock quotes, fund management, insurance companies, credit unions etc.
18	Gambling	Sites that offer or are related to online gambling, lottery, casinos and betting agencies involving chance.	Websites related to gambling, lotteries, casinos, and bookmakers.
19	Games	Sites relating to computer or other games, information about game producers, or how to obtain cheat codes. Game-related publication sites.	Websites related to computer games or other types of games. Resources containing information about game developers, codes, and cheats.
20	Government	Sites run by governmental or military organizations, departments, or agencies, including police departments, fire departments, customs bureaus, emergency services, civil defense, counterterrorism organizations and hospitals.	Websites operated by government or military organizations, agencies or institutions, including police departments, fire departments, customs bureaus, emergency services, civil defense, counter terrorism organizations, and hospitals.
21	Hate and Intolerance	Sites that promote a supremacist political agenda, encouraging oppression of people or groups of people based on their race, religion, gender,	Internet resources dedicated to the issue of encouraging the infringement of people's rights based on race, religion, gender, age, disability,

ID	Name	Description	Description
		age, disability, sexual orientation or nationality.	sexual orientation, or nationality.
22	Health and Medicine	Sites containing information pertaining to health, healthcare services, fitness and wellbeing, including information about medical equipment, hospitals, drugstores, nursing, medicine, procedures, prescription medications, etc	Websites containing information related to health care activities, medical services, fitness, and well-being. This category also includes Internet resources related to medical equipment, hospitals, pharmacies, nursing, medicine, drug prescriptions etc.
23	Illegal Drug	Sites with information on the purchase, manufacture, and use of illegal or recreational drugs and their paraphernalia, and misuse of prescription drugs and other compounds.	Websites containing information on buying, production, and use of illegal or recreational drugs, as well as misuse of prescription drugs.
24	Job Search	Sites containing job listings, career information, assistance with job searches (such as resume writing, interviewing tips, etc.), employment agencies or head hunters.	Websites containing lists of job offerings and providing assistance in finding work. This category includes websites of employment agencies and recruitment companies.
25	Military	Information on military branches,	Information about military

ID	Name	Description	Description
		armed services, and military history.	organizations and military history.
26	Streaming Media and Downloads	Sites that deliver streaming content, such as Internet radio, Internet TV or MP3 and live or archived media download sites. Includes fan sites, or official sites run by musicians, bands, or record labels.	Websites containing streaming content, such as Internet radio, Internet TV, or MP3 broadcasting. This category includes fan pages or official websites of musicians, bands, or record companies.
27	News	Sites covering news and current events such as newspapers, newswire services, personalized news services, broadcasting sites, and magazines.	News websites: online magazines, news feeds etc.
28	Non-profits and NGOs	Sites devoted to clubs, communities, unions, and non-profit organizations. Many of these groups exist for educational or charitable purposes.	Websites dedicated to clubs, communities, unions, and non-profit organizations. Significant number of these organizations were created for educational or charitable purposes.
29	Nudity	Sites that contain full or partial nudity that are not necessarily overtly sexual in intent. Includes sites that advertise or sell lingerie, intimate	Resources containing images of partial or full nudity. This category includes sites that advertise or sell lingerie,

ID	Name	Description	Description
		apparel, or swimwear.	intimate apparel, or swimwear.
30	Personal Sites	Sites about or hosted by personal individuals, including those hosted on commercial sites such as Blogger, AOL, etc.	Personal websites created by individuals or groups of individuals, blogs.
31	Phishing and Fraud	Sites that are used for deceptive or fraudulent purposes (e.g. phishing), such as stealing financial or other user account information. These sites are most often designed to appear as legitimate sites in order to mislead users into entering their credentials	Websites used for deception or fraud (for example, for phishing), such as stealing financial or other information through a user's account. These sites are often cleverly designed copies of popular legitimate resources, which misleads users.
32	Politics	Sites that promote political parties or political advocacy, or provide information about political parties, interest groups, elections, legislation or lobbying. Also includes sites that offer legal information and advice.	Websites dedicated to political parties or engaged in political propaganda and providing information about interest groups, elections, legislation, or lobbying. This category includes websites providing information about rights and their protection.
33	Pornography/ Sexually Explicit	Sites that contain explicit sexual content. Includes adult products such as sex toys,	Websites containing sexually explicit content. This category includes adult

ID	Name	Description	Description
		CD-ROMs, and videos, adult services such as videoconferencing, escort services, and strip clubs, erotic stories and textual descriptions of sexual acts.	products such as sex toys, CDs, and videos. Websites from this category allow users to order escort services, get information about strip clubs, read erotic stories and textual descriptions of sexual acts.
34	Real Estate	Sites relating to commercial or residential real estate services, including renting, purchasing, selling or financing homes, offices, etc.	Information about renting, buying, or selling real estate or land. Recommendations for buying or selling residential properties. Real estate agencies, rental services, relocation services, and housing improvement services.
35	Religion	Sites that deal with faith, human spirituality or religious beliefs, including sites of churches, synagogues, mosques and other houses of worship.	Websites dedicated to faith, spirituality and religious beliefs, including websites of churches, synagogues, mosques, and other houses of worship.

ID	Name	Description	Description
36	Restaurants and Dining	Sites that list, review, promote or advertise food, dining or catering services. Includes sites for recipes sites, cooking instruction and tips, food products, and wine advisors.	Websites containing lists and reviews of catering establishments. This category includes websites containing recipes, instructions and tips, food products, and wine selection recommendations.
37	Search Engines and Portals	Sites enabling the searching of the Web, newsgroups, images, directories, and other online content. Includes portal and directory sites such as white/ yellow pages.	Websites that allow you to search the Internet for news, images, directories, and other types of online content. This category includes portals and website catalogs, such as white/yellow pages.
38	Shopping	Sites for online shopping, catalogs, online ordering, auctions, classified ads. Excludes shopping for products and services exclusively covered by another category such as health&medicine.	E-commerce websites, catalogs, auctions, advertisements. This category does not include resources for goods and services from more specific categories, such as healthcare and medicine.
39	Social Networking	Sites that enable social networking for online communities of various topics, for friendship, dating, or professional reasons.	Social networking websites, where users can communicate, interact, and send messages and files to each other.

ID	Name	Description	Description
40	Spam Sites	Sites that have been promoted through spam techniques.	Websites that are promoted using spam mailings.
41	Sports	Sites relating to sports teams, fan clubs, scores and sports news. Relates to all sports, whether professional or recreational.	International and national websites of teams, sport associations, and sport colleges. Resources containing game results and schedules, sport magazines, and newspapers, virtual sport and sport leagues.
42	Malware	Sites that install unwanted software on a user's computer with the intent to make system changes or enable third-party monitoring without the user's consent.	Websites containing unwanted software that is downloaded and installed on user computers automatically. Viruses make changes to the system and allow attackers to obtain data from the computer without the knowledge and consent of its owner.
43	Stock trading	Promotion and facilitation of securities trading and management of investment assets. Also includes information on financial investment strategies, quotes, and news.	Websites for stock trading and managing investment assets. Also include information on financial and investment strategies, quotes, and news.

ID	Name	Description	Description
44	Translators	Sites that translate Web pages or phrases from one language to another. These sites bypass the proxy server, presenting the risk that unauthorized content may be accessed, similar to using an anonymizer	Websites that allow users to translate web pages to other languages online. These resources can be used to bypass certain restrictions because the URL of the online translator is used to get access to a website.
45	Travel	Sites that provide travel and tourism information or online booking or travel services such as airlines, accommodations, car rentals. Includes regional or city information sites.	Airlines, flight booking agencies, travel planning, advance reservations, vehicle rentals, destination descriptions, hotel, and casino advertising.
46	Violence	Sites that contain images or text depicting or advocating physical assault against humans, animals, or institutions. Sites of a particularly gruesome nature. Sites that contain profanity.	Websites containing images, text descriptions, and propaganda related to physical violence against people, animals, or organizations/ societies. This category includes resources containing illustration of natural disasters and websites containing profanities.
47	Weapons	Sites that depict, sell, review or describe guns and weapons, including for sport.	Websites containing images, reviews, and descriptions of weapons. This category also

ID	Name	Description	Description
			includes e-commerce websites for guns and weapons, including for sports activities.
48	Web-based Email	Sites that enables users to send and receive email through a web-accessible email account.	Portals allowing users to create mail accounts for writing, reading, sending, and receiving emails.
49	General	Sites that do not clearly fall into other categories, for example, blank web pages	Websites that do not fall into other categories, for example, empty web pages.
50	Leisure and Recreation	Sites relating to recreational activities and hobbies including zoos, public recreation centers, pools, amusement parks, and hobbies such as gardening, literature, arts&crafts, home improvement, home décor, family, etc.	Websites related to recreational activities and hobbies, including resources dedicated to zoos, public recreational centers, swimming pools, amusement parks, and hobbies such as gardening, literature, arts and crafts, home decor etc.
51	Online training and tools	Distance education, online courses, vocational training, software training, skills training.	Remote learning, online courses, occupational training, training programs, skill improvement.
52	Legal	Legal websites, law firms, discussions and analysis of legal issues.	Websites of law firms, discussion, and analysis of legal issues.
53	Local Information	City guides and tourist information, including	Travel guides and information for tourists, including

ID	Name	Description	Description
		restaurants, area information and local points of interest.	information on restaurants and local sights.
54	Reference and Research	Personal, professional, or educational reference material, including online dictionaries, maps, library catalogues and scientific information.	Professional or educational reference materials, including online dictionaries, maps, tabulation of census data, almanacs, library catalogs, genealogies, and scientific information.
55	Technical or business forums and news groups	Websites with discussions of user-generated content related to business or technical development.	Websites with discussions related to business or technologies.
56	Technical information and documentation	Websites that provide information on technical information and documentation.	Sites with technical documentation and information.
57	Personal Storage	Websites that allow the uploading of files for remote data storage.	Websites which allow users to upload files and backups for remote data storage.
58	CDNs	Content distribution networks	Content delivery networks
59	Profanity	Websites which contain excessive use of profanity or obscenities.	Websites containing profanities

ID	Name	Description	Description
60	Professional social networks	The subset of social networking websites which includes content intended exclusively for businesses or professionals.	A subset of social media websites that includes content aimed exclusively at businesses and professionals.
61	Botnets	Sites that manage networks of bots through command-and-control centers.	Websites that control bot networks via command and control centers.
62	Cults	Sites relating to non-traditional religious practice typically known as cults, that is, considered to be false, unorthodox, extremist, or coercive, with members often living under the direction of a charismatic leader.	Websites related to unconventional religious practices (cults). Such movements are considered false, unorthodox, and extremist. Cult members often live under the guidance of a charismatic leader.
63	Fashion and Beauty	Sites concerning fashion, jewelry, glamour, beauty, modeling, cosmetics or related products or services. Includes product reviews, comparisons, and general consumer information.	Websites related to fashion, jewelry, glamor, beauty, modeling business, cosmetics and goods or services related to any of the listed above. Include reviews, comparisons, and general information that is useful to customers.
64	Greeting cards	Sites that allow people to send and receive greeting cards and postcards.	Websites allowing users to send and receive greeting cards.

ID	Name	Description	Description
65	Hacking	Sites that promote or give advice about how to gain unauthorized access to proprietary computer systems, for the purpose of stealing information, perpetrating fraud, creating viruses, or committing other illegal activity related to theft of digital inform	Websites that promote or contain recommendations on how to obtain unauthorized access to computer systems for stealing information and fraud, instructions on creating viruses, or other illegal activities related to stealing digital information.
66	Cryptocurrency Mining	Sites that use cryptocurrency mining technology without user permission.	Websites using cryptocurrency mining technologies without user permission.
67	Illegal Software	Sites that illegally distribute software or copyrighted materials such as movies or music, software cracks, illicit serial numbers, illegal license key generators.	Websites that unlawfully distribute software or materials protected by the copyright, such us movies, music, illegal serial numbers, key generators.
68	Image Sharing	Sites that host digital photographs and images, online photo albums and digital photo exchanges.	Portals for uploading digital photographs and images, photo albums, and photo sharing websites.

ID	Name	Description	Description
69	Information Security	Sites that provide legitimate information about data protection, including newly discovered vulnerabilities and how to block them	Websites providing information on legal ways of protecting data, including new threats and ways of dealing with them.
70	Instant Messaging	Sites that enable logging in to instant messaging services such as ICQ, AOL Instant Messenger, IRC, MSN, Jabber, Yahoo Messenger, and the like	Websites allowing users to create accounts on instant messaging services, such as ICQ, AOL Instant Messenger, IRC, MSN, Jabber, Yahoo Messenger etc.
71	Network Errors	Sites that do not resolve to any IP address.	Websites without assigned IP addresses.
72	Parked Domains	Sites that are inactive, typically reserved for later use. They most often do not contain their own content, may simply say "under construction," "purchase this domain," or display advertisements.	Parked domains include URL addresses with restricted content and endpoints for transitions from advertisements. Such resources can generate profit, but usually do not have useful information for users. This category also includes websites under reconstruction, as well as folder and basic pages of web servers.

ID	Name	Description	Description
73	Peer-to-Peer	Sites that enable direct exchange of files between users without dependence on a central server.	Websites allowing users to exchange files directly without the central server.
74	Private IP Addresses	Sites that are private IP addresses as defined in RFC 1918, that is, hosts that do not require access to hosts in other enterprises (or require just limited access) and whose IP address may be ambiguous between enterprises but are well defined within a certain enterprise.	Includes IP address ranges defined in RFC 1918.
75	School Cheating	Sites that promote unethical practices such as cheating or plagiarism by providing test answers, written essays, research papers, or term papers.	Websites containing answers for educational tests, essays, reports, research papers or term papers.
76	Sex Education	Sites relating to sex education, including subjects such as respect for partner, abortion, gay and lesbian lifestyle, contraceptives, sexually transmitted diseases, and pregnancy.	Websites related to sex education, including topics such as respect for your partner, abortion, bisexual lifestyle, contraception, sexually transmitted diseases, and pregnancy.
77	Tasteless	Sites with offensive or	Websites with offensive content,

ID	Name	Description	Description
		tasteless content such as bathroom humor, or gruesome or even frightening content such as shocking depictions of blood or wounds, or cruel animal treatment.	such as obscene humor, dreadful or scary content, shocking images of blood, wounds, or animal abuse.
78	Child Abuse Images	Sites that portray or discuss children in sexual or other abusive acts	Websites containing photographs illustrating child abuse.
79	Gay, Lesbian or Bisexual	Web pages that cater to or discuss the gay, lesbian, bisexual or transgender lifestyle.	Websites containing materials about gays, lesbians, and transgender people.
80	Literature and Books	Web pages for published writings including fiction and non-fiction novels, poems and biographies.	Websites containing published books, novels, short stories (including science fiction), biographies.
81	Nutrition and Diet	Web pages on losing weight and eating healthy, diet plans, weight loss programs and food allergies.	Websites dedicated to proper nutrition, weight loss, and diets.
82	Pets and Animals	Web pages with information or products and services for pets and other animals including birds, fish, and insects.	Websites dedicated to products and services for pets and other animals.
84	Updates	All sites with software updates,	Websites containing software and OS

ID	Name	Description	Description
		OS updates, security updates.	updates and other updates
97	Reputation: Low risk	Sites with low reputational risk. Web sites with no known security risks, but belonging to categories of sites that may be a risk.	Websites with low reputational risk. Usually these are websites, which do not contain known vulnerabilities, but belong to categories where the risk of a security breach is possible.
98	Reputation: Medium risk	Sites with medium reputational risk. Web sites that may be harmless, but belong to categories of sites that could harm your computer.	Websites with middle reputational risk. Usually these are websites, which may contain vulnerabilities and which belong to categories where the risk of a security breach is possible.
99	Reputation: High risk	Sites with high reputational risk. Web sites that are confirmed to be a security risk.	Websites with high reputational risk. Websites with known vulnerabilities.

List of supported HTTP headers

HTTP Header	Request/ Response	SET/REPLACE/ ENCRYPT	APPEND	DELETE
Accept	Request	✅	✅	✅
Accept-Charset	Request	✅	✅	✅
Accept-Encoding	Request	✅	✅	✅

HTTP Header	Request/ Response	SET/REPLACE/ ENCRYPT	APPEND	DELETE
Accept-Language	Request	✓	✓	✓
Accept-Ranges	Response	✓	✓	✓
Age	Response			
Allow	Request/ Response	✓	✓	✓
Authorization	Request			
Cache-Control	Request/ Response	✓	✓	✓
Client-IP	Request	✓	✓	
Connection	Request/ Response		✓	
Content-Encoding	Request/ Response		✓	
Content-Language	Request/ Response		✓	
Content-Length	Request/ Response			
Content-Location	Request/ Response		✓	✓
Content-Range	Request/ Response			
Content-Type	Request/ Response			
Cookie	Request	✓	✓	✓
Date	Request/ Response			
ETag	Response	✓	✓	
Expect	Request	✓		

HTTP Header	Request/ Response	SET/REPLACE/ ENCRYPT	APPEND	DELETE
Expires	Request/ Response		✓	✓
From	Request	✓	✓	
Host	Request			
If-Match	Request	✓		
If-Modified-Since	Request			
If-None-Match	Request	✓		
If-Range	Request			
If-Unmodified-Since	Request			
Last-Modified	Request/ Response			
Location	Response	✓	✓	
Max-Forwards	Request			
Meter	Request/ Response		✓	✓
Pragma	Request/ Response		✓	✓
Proxy-Authenticate	Response	✓		
Proxy-Authorization	Request	✓		
Proxy-Connection	Request	✓		
Range	Request	✓	✓	
Referer	Request	✓	✓	
Retry-After	Response	✓	✓	
Server	Response	✓	✓	
Set-Cookie	Response	✓	✓	✓

HTTP Header	Request/ Response	SET/REPLACE/ ENCRYPT	APPEND	DELETE
TE	Request	✓		
Trailer	Request/ Response		✓	
Transfer-Encoding	Request/ Response		✓	
Upgrade	Request/ Response		✓	
User-Agent	Request	✓	✓	
Vary	Response	✓	✓	✓
Via	Request/ Response	✓	✓	✓
Warning	Request/ Response	✓	✓	✓
WWW-Authenticate	Response			

PLATFORM MANAGEMENT CONTROLLER COMMAND LINE INTERFACE (PMC CLI)

General Information

For UserGate hardware and software systems (HSC) equipped with the PMC (Platform Management Controller) control module, the PMC CLI command line interface is available. Using interface commands, you can view the status of hardware platform elements, monitor the platform's operation, manage its settings, and access to it.

You can connect to the PMC via the console port or the MGMT port located on the UserGate device panel. For more information on the PMC module availability on a

device and the types of available connection ports, see the Hardware platforms section of the documentation.

The PMC CLI has two main operating modes: the **bootloader mode** and the **main PMC software mode**.

To enter the PMC CLI, press the **Enter** key while the following line is displayed in the device console:

```
Hit 'Enter' key to stop autoboot: 3
```

The **bootloader mode** (PMC loader) is designed to ensure a hardware and software system can be restored to working order. This mode is only available when using console port connection.

The bootloader mode prompt looks like this:

```
loader>
```

The loader has an inactivity timer running in the CLI. That is, if no commands are entered within 45 seconds, the device will reboot.

To access the **main PMC software mode**, you need to enter your login and password. Pressing **Ctrl +]** will resume booting the device in a standard mode:

```
Press '^]' for autoboot and connect to aux
PMC login:
```

After three unsuccessful login attempts, a mandatory 10-second timeout takes place before each subsequent attempt. During the timeout, you can reset the PMC settings to the original ones using the key combination **Ctrl + e**.

```
PMC login: 111
Password:
You have exceeded the number of attempts, 10 seconds timeout...

All configuration will be cleared and device will be reset. Continue?
(y/n):
```

The default login/password in PMC CLI is **admin**/**password**.

The main PMC software mode has two internal modes: a viewing mode and a configuration mode.

After a user has logged in to the console, the PMC CLI viewing mode is activated. In this mode, commands for viewing the status and commands for running some utilities can be executed. Device control and configuration commands are not available in this mode.

The prompt in the command line in the PMC CLI viewing mode looks as follows: "**>**".

```
PMC>
```

To enter the PMC CLI device configuration mode, use the **configure** command. In the configuration mode, the command line prompt changes to "**#**".

```
PMC> configure
PMC#
PMC# exit
PMC>
```

Built-in help

Pressing the **Tab** key at any point while entering a command displays a list of possible command continuation options with a brief description.
Examples:

```
PMC>
+ history          Display the history list
+ reset            Reset peripheral sub-system or board
+ autoboot         Run autoboot command
+ traceroute       Print the route to network host using ICMP
+ aux              AUX terminal support
+ version          Show version
+ show             Show parameter
+ exit             Logout/Save config and exit
+ configure        Configuration mode
```

```
+ debug            Debug mode
+ ping             Send ICMP ECHO_REQUEST to network host
```

```
PMC> au
+ autoboot         Display the history list
+ aux              Factory autotest support
```

```
PMC# show
+ network          Network-settings sub-system
+ date             pmc date and time settings
+ uptime           pmc uptime
+ users            Users-settings sub-system
+ factory          Factory-settings sub-system (Type, S/N, MAC)
+ platform         Platform-settings sub-system
```

```
PMC# show network s
+ ssh              SSH server settings
+ settings         Network settings
+ status           Network status
```

Command structure

All configuration commands in the CLI have the following structure:

```
<action> <level> <filter> <configuration_info>
```

where:

<action>: the action to be performed (*create, set, show, delete*).

<level> is the configuration level (*cli, platform, network, factory, users*).

<filter> is the identifier of the object being accessed.

<configuration_info> is the set of parameter values to be applied to the *<filter>* object.

Example

```
PMC> set network gateway 192.168.1.1
```

There are also commands which allow to perform actions not related to configuration. These commands have the following structure:

```
<util> <util_parameters>
```

where:

<util>: actions not related to device configuration:

: additional parameters of performed actions.

Example

```
PMC> ping 192.168.1.1
```

Show

The **show** command displays all values at the specified level, and everything deeper.

Example

```
PMC> show platform
------------------
Bypass map:
        Relay 1: port0 <-> port1
        Relay 2: port2 <-> port3
        Relay 3: port4 <-> port5
        Relay 4: port6 <-> port7
Bypass state:
        Relay 1: DISABLED
        Relay 2: DISABLED
        Relay 3: DISABLED
        Relay 4: DISABLED
------------------
Fan state:
        Mode:   auto
```

```
        Level:  255
        Speed:  0 RPM
------------------
SoC is Stopped
------------------
Power control signals:
    LS1084_IO_PWR: Enabled
    M2_PWR:     Enabled
    CORE_PWR:   Enabled
    DDR4_PWR:   Enabled
Power Inputs:
    POW_IN1:    None
    POW_IN2:    OK
PG signals:
    LS1084_IO_PWR_PG_E: OK
    CORE_PWR_PG_E:      OK
    DDR4_PWR_PG_E:      OK
    M2_PWR_PG_E:        OK
    USB_PWR_PG_E:       OK
    FPGA_PWR_PG_E:      OK
Voltages:
     VDD: 1.030V - OK
    SVDD: 1.018V - OK
    XVDD: 1.414V - OK
    DVDD: 3.378V - OK
    OVDD: 1.843V - OK
    GVDD: 1.257V - OK
    FPGA: 3.434V - OK
Currents:
     VDD: 2.668A - OK
    GVDD: 0.195A - OK
------------------
Temperature:
        Board               :  41.000 C
        CPU                  :  44.500 C
        ALERT signal        : Inactive
------------------
```

If you only need temperature information:

```
PMC> show platform therm
Temperature:
        Board                :  41.000 C
        CPU                   :  44.500 C
        ALERT signal         : Inactive
```

If you only need temperature values:

```
PMC> show platform therm value
Temperature:
        Board:  41.000 C
          CPU:  43.500 C
```

Telemetry output

Readings from some sensors are collected in logs. Logs are stored for the last 60 seconds, 60 minutes and 72 hours. Such information can be output in two formats: a graphical (graph), and a textual (log).
Here is an example of telemetry output in the graphical format:

```
PMC> show platform power graph
    3333233333333333333333233333233333333322223322323323332333333
    2222931133332232241000801115823330444188884177039112900231222
    8886323325224294435966995770222274661797701588058118899405666
    6664793391992042285088405773909936006742276555311226100275444
3600
3510                  ▒           ▒     ▒▒
3420          ▒   ▒   ▒           ▒   ▒ ▒▒▒     ▒   ▒
3330▒▒▒▒ ▒  ▒▒▒▒▒ ▒▒▒▒       ▒   ▒▒▒ ▒▒▒     ▒   ▒   ▒   ▒▒ ▒▒
3240▒▒▒▒ ▒  ▒▒▒▒▒▒▒▒▒▒     ▒▒▒▒ ▒▒▒▒ ▒▒▒▒    ▒▒   ▒   ▒   ▒▒▒▒▒
3150▒▒▒▒ ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒▒ ▒▒▒▒    ▒▒   ▒ ▒▒▒  ▒▒▒▒▒▒▒
3060▒▒▒▒ ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒▒▒▒▒▒    ▒▒   ▒▒▒▒▒▒▒▒▒▒▒▒▒
2970▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▒▒▒▒▒▒▒▒▒   ▒▒   ▒▒▒▒▒▒▒▒▒▒▒▒▒
2880▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒  ▒▒▒▒▒▒▒▒▒▒▒▒▒
2790▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
2700▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
    0....5....1....1....2....2....3....3....4....4....5....5....
              0    5    0    5    0    5    0    5    0    5
       CPU current, mA per 1 second(s) (last 60 seconds)
            ▒ = maximum value; ▒ = average value
    4443333333333333333333333333333333
    7107755544544544345555555344343455
    8704269413086636734444220761515300
    7843190768320980387777553306161833
4800▒
4620▒
4440▒
4260▒▒
4080▒▒▒
3900▒▒▒▒▒
3720▒▒▒▒▒▒▒     ▒    ▒▒▒▒
3540▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▒ ▒▒▒▒
3360▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
3180▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
3000▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
    0....5....1....1....2....2....3....3....4....4....5....5....
              0    5    0    5    0    5    0    5    0    5
          CPU current, mA per minute (last 60 minutes)
            ▒ = maximum value; ▒ = average value
```

Example of telemetry output in text format:

```
PMC> show platform power log
CPU current, mA for last 60 recorded second(s)
+--------------------+---------+---------+
|               time | maximum | average |
+--------------------+---------+---------+
| 07.12.2023 17:04:43 |    3242 |    3242 |
| 07.12.2023 17:04:44 |    3264 |    3264 |
...
| 07.12.2023 17:05:42 |    3242 |    3242 |
+--------------------+---------+---------+
CPU current, mA for last 35 recorded minute(s)
+--------------------+---------+---------+
|               time | maximum | average |
+--------------------+---------+---------+
| 07.12.2023 16:31:16 |    4787 |    4113 |
| 07.12.2023 16:32:16 |    4178 |    3857 |
...
| 07.12.2023 17:05:16 |    3590 |    3208 |
+--------------------+---------+---------+
CPU current, mA for last 0 recorded hour(s)
+--------------------+---------+---------+
|               time | maximum | average |
+--------------------+---------+---------+
+--------------------+---------+---------+
```

You can display a table or graph only for a certain period: **minute**/**hour**/**day**:

```
PMC> show platform soc mem graph hour
```

Execute Commands

The following commands are available in this section:

- autoboot;
- aux;
- history;
- ping;
- traceroute;
- reset;
- update;
- version;
- configure;

diff;

- revert;
- exit.

autoboot

Running this command in bootloader mode will start the main PMC firmware.

```
loader> autoboot

Try to load primary-image:
Reading 16 bytes (0x10) at offset 0x7000000 ... OK (0 KiB/s)
Reading 270664 bytes (0x42148) at offset 0x7000010 ... OK (87 KiB/s)

## Starting application at 0x24000000 ...
```

Running this command in PMC mode starts the host processor and enables aux mode.
Running the command without parameters launches the main UGOS image; adding the **recovery** parameter launches the recovery image:

```
PMC> autoboot
Connected to CPU at speed 115200.
Escape character is '^]'.

NOTICE:  UDIMM M471A1K43EB1-CWE
NOTICE:  Build-in self test passed

NOTICE:  8 GB DDR4, 64-bit, CL=15, ECC off
NOTICE:  BL2: v2.4(release):
NOTICE:  BL2: Built : 03:40:29, Dec 30 2021
NOTICE:  BL2: Booting BL31
NOTICE:  BL31: v2.4(release):
NOTICE:  BL31: Built : 03:40:29, Dec 30 2021
NOTICE:  Welcome to ls1088ardb BL31 Phase
```

```
U-Boot 2021.04 (Dec 30 2021 - 03:40:29 +0000), Build: test-
p4_ugos_g-380
```

aux

Connecting to the host processor terminal. To return to PMC CLI, press "**Ctrl +]**":

```
PMC> aux
Connected to CPU at speed 115200.
Escape character is '^]'.

UGOS login:
UGOS login: ^]
Disconnected from CPU

PMC>
```

history

Displays a list of previously executed commands:

```
PMC> history
9 history
8 version
7 show network
6 show platform fan
5 show date
4 show factory
3 version
2 aux
1 autoboot
PMC>
```

ping

Sending an ICMP ECHO request. Perform 5 times with an interval of 1 second. The size of the data in the packet can be specified as an optional parameter.
The command can be stopped by pressing the keyboard shortcut **"Ctrl + C"**:

```
PMC> ping 192.168.1.1
40 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0 ms
40 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0 ms
40 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0 ms
40 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0 ms
40 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0 ms
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss
rtt min/avg/max = 0/0/0 ms
PMC>

PMC>
PMC> ping 192.168.1.1 1400
1408 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1 ms
1408 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0 ms
1408 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0 ms
1408 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0 ms
1408 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0 ms
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss
rtt min/avg/max = 0/0/1 ms
PMC>
```

traceroute

Checking the route to the requested IP address:

```
PMC> traceroute 192.168.70.11
traceroute to 192.168.70.11(192.168.70.11) 30 hops max 32(40) bytes of
data
1    192.168.75.1 299 ms 94 ms 4 ms
2    192.168.70.11 0 ms 0 ms 0 ms
```

reset

Resetting the device:

```
PMC> reset

PMC(Loader) Firmware 1.0 build dev (2021-12-27 - 10:01:23)

MCU: dev_id 0x450, rev_id 0x2003
Flash size: 128 KiB
Booting from Flash
Watchdog: Enabled
PMC[PWR]: Power Input 1 in state None
PMC[PWR]: Power Input 2 in state OK
PMC[PWR]: Power supplies are turned off.
Configure Clock Generator (SI5332):
..................OK
PMC[PWR]: Power supplies started.
```

update

Loading software images and security keys into the device's flash memory. Available only in configuration mode.

The firmware update is performed using the command:

```
update firmware <pmc|pmc-backup|boot|sys-recovery> tftp <address>
<filename>
```

To update the SSH server's private key, use the following command:

```
update key ssh tftp <address> <filename>
```

The key should be generated in advance and saved in *.der* format.

To generate a key, you can use the following command:

```
openssl ecparam -genkey -name prime256v1 -noout -outform DER -out
privatekey.der
```

version

Displaying the current PMC firmware version:

```
PMC> version
PMC(Main) Firmware 7.1.0 build 249B (2023-06-07 - 03:55:58)
PMC>
```

Output of the loader version:

```
PMC> version loader
PMC(Loader) Firmware 7.1.3 build 425DEV (2024-08-19 - 23:35:33)
PMC>
```

configure

Entering configuration mode. After executing this command, the prompt in the console changes and it becomes possible to use the commands **set**, **delete**, **create**, **diff**, **revert**. Commands for direct platform management (which are not stored in non-volatile memory) are processed immediately when a command is executed. Settings that require saving are applied and saved only after the **exit** command.

```
PMC> configure
PMC#
PMC# exit
PMC>
```

diff

The command shows the differences between the applied configuration and the one currently installed in configuration mode. Available only in configuration mode:

```
PMC# diff
IP address:
 old: 192.168.75.92 new: 192.168.75.96
Netmask:
 old: 255.255.255.0 new: 255.255.0.0
DHCP:
```

```
 old: off new: on
PMC#
```

revert

Command to reset temporary configuration. Available only in configuration mode:

```
PMC# revert
Temporary configuration revert complete
PMC# diff
Configuration not changed
PMC#
```

exit

In configuration mode, running the command exits configuration mode. If the configuration has been changed, the command applies and saves it:

```
PMC# set network ip 192.168.75.96
Set ip address complete
PMC# exit
New configuration apply complete
PMC>
```

In view mode, running the command exits the terminal (in the case of SSH, it terminates the connection):

```
PMC> exit

Press '^]' for autoboot and connect to aux
PMC login:
```

Commands for Working With Factory Settings

These parameters are set only once during production testing. The following parameters are read only.

To show the parameters:

```
PMC> show factory
```

To view the value of the parameter, specify the parameter:

- sn: device serial number
- type: device type
- mac: physical address (MAC address)

```
PMC> show factory
Factory-settings:
        Device Type: UGAC-101
        S/N:         UGAC21430000015
        Base MAC:    34:91:6F:00:00:7F

PMC>

PMC> show factory mac
34:91:6F:00:00:7F
PMC>
```

Platform Management Commands

Configuration level **platform** is corresponding to this section. Commands at this level allow to manage components of the UserGate device.

Setting up the Bypass Relay

This functionality is not available on all devices.

Management commands are performed at the **platform bypass** level. This section implements managing the state of bypass relays of network ports and viewing the current state and mapping of the relays.

Set command format:

```
PMC> set platform bypass <relay-number> <state>
PMC> set platform bypass all <state>
```

where:

<relay-number>: the number of the relay which is configured.

<state>: the state of the relay. Available values:

- **enable**: global setting; it allows to use relay functionality. When the power is off, the relay will be closed. The value is stored in non-volatile memory (if the relay is set to **enable**, then after the device is restarted the relays will remain closed, until the opening command is sent).
- **disable**: global setting; the value is stored in non-volatile memory (if the relay is set to **disable**, then after the device is restarted the relays will open while PMC is restarted).
- **on**: closing of the bypass relay (traffic is bypassing the processor). The state is reset after the device is restarted.
- **off**: opening of the bypass relay (traffic is bypassing the processor). The state is reset after the device is restarted.

Examples of the commands:

```
PMC> set platform bypass all enable
Enable all bypass relays
PMC> set platform bypass 1 disable
Disable bypass relay 1
PMC> set platform bypass 3 off
Set bypass relay 3 state OFF
PMC>
```

Example output of the state view command:

```
PMC> show platform bypass
Bypass map:
        Relay 1: port0 <-> port1
        Relay 2: port2 <-> port3
        Relay 3: port4 <-> port5
```

```
        Relay 4: port6 <-> port7
Bypass state:
        Relay 1: DISABLED
        Relay 2: ON
        Relay 3: OFF
        Relay 4: ON
PMC>
```

Cooling fan management

This section is located at the **platform fan** level. The commands in this section allow to control the cooling fan operating mode and to see their current status.

The fans have two operating modes: manual and automatic. By default the fan operates the in automatic mode. The fan speed is controlled by the CPU temperature.

To change the fan operating mode, use the following command:

```
set platform fan mode <auto|manual>
```

In manual mode you can set the fan speed in percent:

```
set platform fan speed <0-100>
```

If there is more than one fan in the platform, you can control the speed of each fan separately in the manual mode.

The cooling fan settings are stored in the non-volatile memory and are not reset after the system is rebooted.

Settings examples:

```
PMC> configure
PMC# set platform fan mode manual
Fan control will be switched to manual mode. These settings will be
saved even after the device is rebooted.Do you really want to switch
the fan control mode (y/n): y
set fan manual mode ok
PMC# set platform fan speed 50 0
```

```
set fan speed ok
PMC# show platform fan
Fan config:
    direction            : front-to-back
Fan state:
    Mode                 : manual
Fan 0:
    Present              : Yes
    Level                : 50%
    Direction            : front-to-back
    Speed                : 17024 RPM
    Alarm                : No
Fan 1:
    Present              : Yes
    Level                : 25%
    Direction            : front-to-back
    Speed                : 10848 RPM
    Alarm                : No
Fan 2:
    Present              : Yes
    Level                : 25%
    Direction            : front-to-back
    Speed                : 10904 RPM
    Alarm                : No
```

Some platform support installing a different direction cooling system. The fan direction is determined automatically after the previous value is removed:

```
PMC# delete platform fan direction
Fan direction reset is complete.
Save config and restart device for autodetect fan direction.
PMC# exit
New configuration apply complete
PMC> reset
Device will be reset. Continue? (y/n): y
```

To show the current state, use the following command:

```
PMC> show platform fan [status|graph|log]
```

```
PMC> show platform fan
Fan state:
        Mode:   manual
        Level:  40
        Speed:  6029 RPM
PMC>
```

The **graph** and **log** parameters refer to the telemetry output.

Starting/Stopping the Host Processor, Viewing Saved Logs

Command to start or stop the processor:

```
set platform soc <start|stop>
```

Command to view processor status:

```
show platform soc [log|load|frequency|mem]
```

The **load**/**frequency**/**mem** parameters refer to the telemetry output.

Example output:

```
PMC> show platform soc
SoC is Running
System time: 08.06.2023 02:47:35
System uptime: 0 day(s), 0 hour(s), 12 minute(s)
Load average: 0.52 0.36 0.31
MemTotal: 61218 MiB
MemFree: 55064 MiB
CPU Frequency: 2200 MHz (Max 2200 MHz)
```

The format of the command to view the last CPU output is:

```
PMC> show platform soc log
NOTICE:  UDIMM M471A1K43EB1-CWE
...
Server is starting, please wait...
Please press Enter to activate this console.
[   15.495488] kmodloader: loading kernel modules from /etc/modules.d/*
[   15.506171] kmodloader: done loading kernel modules from /etc/
modules.d/*
[   15.577127] random: crng init done
[   15.580535] random: 4 urandom warning(s) missed due to ratelimiting
[   22.759147] IPv6: ADDRCONF(NETDEV_UP): port0: link is not ready
```

Temperature monitoring

Temperatures of the host processor and of the board are monitored at the **platform therm** level.

To view the current status, use the following command:

```
PMC> show platform therm [value|status|graph|log]
```

The **graph** and **log** parameters refer to the telemetry output.

To view the temperatures of the host processor and the board only, use the following command:

```
PMC> show platform therm value
```

Example

```
PMC> show platform therm
Temperature:
        Board                 :  58.000 C
        CPU                   :  68.750 C
        ALERT signal          : Inactive PMC>
PMC> show platform therm value
Temperature:
        Board:  57.000 C
```

```
        CPU:    69.000 C
PMC>
```

Power Supply Monitoring

The section of power source monitoring is located at the **platform power** level. To view the information on the working state of power sources, use the following command:

```
show platform power [status|measurements|graph|log]
```

The **graph** and **log** parameters refer to the telemetry output.

Example

```
PMC> show platform power
Power control signals:
    LS1084_IO_PWR: Enabled
    M2_PWR:     Enabled
    CORE_PWR:   Enabled
    DDR4_PWR:   Enabled
Power Inputs:
    POW_IN1:    None
    POW_IN2:    OK
PG signals:
    LS1084_IO_PWR_PG_E: OK
    CORE_PWR_PG_E:      OK
    DDR4_PWR_PG_E:      OK
    M2_PWR_PG_E:        OK
    USB_PWR_PG_E:       OK
    FPGA_PWR_PG_E:      OK
Voltages:
     VDD: 1.039V - OK
    SVDD: 1.012V - OK
    XVDD: 1.401V - OK
    DVDD: 3.382V - OK
    OVDD: 1.845V - OK
    GVDD: 1.245V - OK
    FPGA: 3.404V - OK
```

```
Currents:
     VDD: 7.006A - OK
    GVDD: 0.473A - OK
PMC>
PMC> show platform power measurements
Voltages:
     VDD: 1.040V - OK
    SVDD: 1.013V - OK
    XVDD: 1.403V - OK
    DVDD: 3.384V - OK
    OVDD: 1.846V - OK
    GVDD: 1.248V - OK
    FPGA: 3.404V - OK
Currents:
     VDD: 7.018A - OK
    GVDD: 0.516A - OK
PMC>
```

Command to turn on/off all power except for backup power, i. e. PMC keeps working:

```
PMC> set platform power <on | off>
```

Turning off:

```
PMC> set platform power off
PMC[Netcard]: Power supplies are turned off.
PMC[PWR]: Power supplies are turned off.
```

Turning on:

```
PMC> set platform power on
PMC[PWR]: Signal 'V3V3_PG_E' ready
PMC[PWR]: Signal 'KSZ_PG_E' ready
PMC[PWR]: Signal 'OVDD_PG_E' ready
PMC[PWR]: Signal 'USB_5V0_PG_E' ready
PMC[PWR]: Signal 'USB_SVDD_PG_E' ready
```

```
Configure Clock Generator (5P49V6901):
.........................OK
PMC[PWR]: Signal 'SVDD_PG_E' ready
PMC[PWR]: Signal 'GVDD_PG_E' ready
PMC[PWR]: Signal 'TA_BB_PG_N_E' ready
PMC[Netcard]: Power supplies started.
Configure Clock Generator (CDCM6208V2):
......OK
Configure Clock Generator (CDCM6208V2):
......OK
Configure Clock Generator (CDCM6208V2):
......OK
PMC[PWR]: Power supplies started.
PMC>
```

Alarm Monitoring

This section is located at the **platform alarm** level.
This section of the PMC CLI allows you to view and clear the alarm log and control the alarm sound (on older platforms, starting with the D250).

Alarm log output:

```
PMC> show platform alarm log
01.07.2023 14:12:07 [POWER] VDD: Over/Under voltage condition detected
03.07.2023 11:13:14 [POWER] AVDD: Power good signal missing
05.07.2023 09:16:45 [POWER] GVDD: Over current condition detected
10.07.2023 15:16:47 [POWER] Power input 1: Present signal missing
12.07.2023 20:58:32 [POWER] Power input 2: Vin under voltage condition
detected
15.07.2023 03:41:27 [POWER] Power input 2: Vout over/under voltage
condition detected
15.07.2023 21:14:48 [POWER] Power input 1: Over current condition
detected
18.07.2023 10:52:38 [POWER] Power input 2: Over heat condition detected
19.07.2023 02:29:46 [POWER] Power input 1: Fan fault condition detected
19.07.2023 23:09:56 [THERM] FPGA/SoC/PHY: Over heat condition detected
20.07.2023 06:14:31 [THERM] Common: Over heat shutdown occurred
21.07.2023 15:35:27 [THERM] FAN 1: Present signal missing
```

```
23.07.2023 21:59:31 [THERM] FAN 2: Wrong direction detected
25.07.2023 16:51:46 [THERM] FAN 3: Over current condition detected
27.07.2023 17:36:17 [THERM] FAN 4: Tachometer signal missing
29.07.2023 12:31:24 [THERM] FAN 5: Tachometer readings do not match the
control signal
30.07.2023 10:39:54 [SYS] SoC: Main processor not respond
30.07.2023 19:48:26 [SYS] SoC: Main processor rebooted
31.07.2023 14:51:55 [SYS] SoC: High CPU load detected
31.07.2023 19:25:38 [SYS] SoC: High memory usage detected
```

Displaying the global alarm status:

```
PMC# show platform alarm status
Alarm disabled
```

Clearing the alarm log:

```
PMC# delete platform alarm log
Alarm log cleared
```

.Alarm sound control (not available on all platforms):

```
PMC# set platform alarm on
Alarm enabled
PMC# set platform alarm off
Alarm disabled
```

Commands for Managing Network Settings

The commands for network settings management are available at the **network** level. In this section it is possible to set and view the following parameters:

- dhcp: turning the DHCP client on/off.
- ip: the static IP address of the device.
- netmask: the network mask of the device.

- gateway: the default gateway.
- vlan: the VLAN ID of the device.
- arp: the static and the dynamic ARP records.
- ssh-port: the port for the incoming SSH server connections.
- nameserver: the DNS server address.
- rule: the access rules for the MGMT port. Managing white and black lists of addresses.

Command to view the current state of the interface:

```
PMC> show network status
-----------------------------------
Network status:
IP address type: static
IP address: 192.168.1.6
Gateway address: 192.168.1.1
Netmask: 255.255.255.0
-----------------------------------
```

To view network settings:

```
PMC> show network
-----------------------------------
Network status:
IP address type: static
IP address: 192.168.1.6
Gateway address: 192.168.1.1
Netmask: 255.255.255.0
-----------------------------------
Network settings:
DHCP: off
IP address: 192.168.1.6
Gateway address: 192.168.1.1
Netmask: 255.255.255.0/24
VLAN id: 0
```

```
SSH server port: 22
DNS address: 0.0.0.0
URL (for UGOS updating): none
-----------------------------------
 Address         HWaddress               Flags
192.168.1.80    34:67:8A:4F:91:10       static
192.168.1.1     50:3E:AA:16:2B:97       dynamic
-----------------------------------
Access rules:
-----------------------------------
Type: ip range
Start address: 192.168.1.1
End address: 192.168.1.10
Access: Drop
-----------------------------------

PMC>
PMC> show network arp
Address         HWaddress               Flags
192.168.1.80    34:67:8A:4F:91:10       static
PMC>
```

Once the values have been set, the network settings are applied and saved to non-volatile memory.

To reset to default values, use the **delete** command:

```
delete network [ip|netmask|gateway|arp|vlan|ssh-port]
```

When running the command without parameters, all network settings are reset to default values:

```
PMC> show network
------------------
IP address: 192.168.1.75
Gateway address: 192.168.1.10
Netmask: 255.255.255.128
VLAN id: 12
SSH server port: 4001
```

```
------------------
Address         HWaddress               Flags
192.168.1.80    34:67:8A:4F:91:10       static
------------------
PMC> delete network vlan
Set default vlan complete
PMC>PMC> show network
------------------
IP address: 192.168.1.75
Gateway address: 192.168.1.10
Netmask: 255.255.255.128
VLAN id: 0
SSH server port: 4001
------------------
Address         HWaddress               Flags
192.168.1.80    34:67:8A:4F:91:10       static
------------------
PMC> delete network
Using default configuration

PMC> show network
------------------
IP address: 192.168.1.2
Gateway address: 192.168.1.1
Netmask: 255.255.255.0
VLAN id: 0
SSH server port: 22
------------------
Address         HWaddress               Flags
------------------
PMC>
PMC> delete network rule
Access rules cleared successfully
```

dhcp

By default, DHCP is off, and the IP address specified in settings is used.

To turn DHCP on, use the following command:

```
PMC> set network dhcp on
DHCP client start complete
PMC>
```

To view the settings received over DHCP, use the following command:

```
PMC> show network status
-----------------------------------
Network status:
IP address type: dhcp
IP address: 192.168.75.238
Gateway address: 192.168.75.1
Netmask: 255.255.255.0
-----------------------------------
PMC>
```

ip

Static IP address, by default it is 192.168.1.2.

Example of a command to set an IP address:

```
PMC> set network ip 192.168.1.75
Set ip complete
PMC> show network ip
IP address: 192.168.1.75
PMC>
```

The IP address can be specified by specifying a subnet mask:

```
PMC> set network ip 192.168.75.88/24
Set ip address complete
PMC> set network ip 192.168.75.88 255.255.255.0
Set ip address complete
PMC>
```

gateway

The default gateway, after resetting the settings, is 192.168.1.1.

To change this value, use the following command:

```
PMC> set network gateway 192.168.1.10
Set gateway complete
PMC> show network gateway
Gateway address: 192.168.1.10
PMC>
```

nameserver

By default DNS server address is not set and is equal to 0.0.0.0.

To change this value, use the following command:

```
PMC> set network nameserver 192.168.1.1
Set nameserver complete
PMC>
```

To view the value, use the following command:

```
PMC> show network nameserver
DNS address: 192.168.1.1
PMC>
```

vlan

By default, the VLAN ID is set to 0.

In this mode the device receives all packets and does not use VLAN tag when sending packets.

If the vlan value is set to a non-zero value, all packets arriving with a tag different from the one set, or without a tag, are filtered.

When sending, the set tag is added to the Ethernet header.

To change this value, use the following command:

```
PMC> set network vlan 12
Set vlan complete
PMC> show network vlan
VLAN id: 12
PMC>
```

arp

The device's arp table is empty by default. When communicating with other network nodes dynamic records are added to the table. These records have a short lifetime and are always removed after a restart.

It is possible to set static entries with *ipaddr*/*hwaddr* pairs. If a destination IP address is present in the table, then when the packet is sent, the device does not perform an arp request, but inserts the MAC address from the table into the Ethernet header.

To add static arp entries, use a command of the following format:

```
set network arp <ip addr> <hw addr>
```

To delete:

```
delete network arp [ip addr]
```

When **delete** is executed without specifying an address, all records are deleted.

Example of the commands:

```
PMC> set network arp 192.168.1.55 34:67:8A:4F:91:55
PMC> show network arp
Address         HWaddress               Flags
192.168.1.80    34:67:8A:4F:91:10       static
192.168.1.65    34:67:8A:4F:91:43       static
192.168.1.55    34:67:8A:4F:91:55       static
PMC> delete network arp 192.168.1.65
PMC> show network arp
Address         HWaddress               Flags
192.168.1.80    34:67:8A:4F:91:10       static
192.168.1.55    34:67:8A:4F:91:55       static
```

```
PMC> delete network arp
PMC> show network arp
Address         HWaddress               Flags
PMC>
```

ssh-port

Port for incoming connections of PMC SSH server. The default value is 22.

To change this value, use the following command:

```
PMC> set network ssh-port 4001
Set ssh-port complete
PMC> show network ssh-port
SSH server port: 4001
PMC>
```

rule

Access rules for MGMT port. They allow to create white and black lists of IP addresses for access to the PMC.

Adding and deleting rules is only possible in configuration mode. The rules are applied when exiting the configuration mode using the <0>exit command.

Examples of adding rules:

```
PMC> set network rule allow 192.168.1.1 192.168.1.20
Access rule added successfully
PMC> set network rule drop 192.168.1.5 192.168.1.10
Access rule added successfully
```

To view a rule, use the following command:

```
PMC> show network rule
Access rules:
-----------------------------------
Type: ip range
Start address: 192.168.1.1
```

```
End address: 192.168.1.20
Access: Allow
-----------------------------------
Type: ip range
Start address: 192.168.1.5
End address: 192.168.1.10
Access: Drop
-----------------------------------
PMC>
```

To remove a rule, use the following command:

```
PMC> delete network rule 192.168.1.1 192.168.1.20
Access rule deleted successfully
PMC>
```

interface

Enable/disable an MGMT interface.

By default the MGMT interface is enabled.

The example when the interface is enabled:

```
set network interface up
MGMT interface up setted
Parameter will be applied after exit from configure mode
PMC# exit
New configuration apply complete
PMC> INFO: PMC[NET]: Link is up
```

Commands for Managing User Settings

The commands for managing user settings are available at the **users** level. In this section the commands for creation and removal of users and setting authentication methods on SSH server are described. The device allows to add up to 10 users.

By default there is one user **admin** with default password **password**.

```
PMC> show users
------------------
User1:
Login: admin
Public key type:
Public key:
Authentication methods: password
------------------
PMC>
```

Creating a User

To create a user, use the following command:

```
PMC> create users <username>
```

Once created, the user must set a password or public key. The authentication method must be set before the user can start working in the system.

Example of user creation:

```
PMC> create users slon
New user slon created
PMC> show users slon
Login: slon
Public key type:
Public key:
Authentication methods: none
PMC>
```

Setting up an authorization method

Available authentication methods include password and ECDSA public key. It is possible to set one or both of these methods for the user at the same time.

To disable one of authentication methods, remove key or password. Use the command with the **delete** action.

Example

```
PMC> set users slon password
Changing password for slon
Enter new password:
Retype new password:
Password for user slon setted
PMC> show users slon
Login: slon
Public key type:
Public key:
Authentication methods: password
PMC>
PMC> set users slon key ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEs+nzq8LrO0OeniYHT
YnTPtbc+CKaLAgvKz+3faiu5qQT7VN1kaPKx1hBV/e3HFzCen3XMVPerkOUEx0q2WrV0=
Public key for user slon setted
PMC>
PMC> show users slon
Login: slon
Public key type: ecdsa-sha2-nistp256
Public key:
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEs+nzq8LrO0OeniYHT
YnTPtbc+CKaLAgvKz+3faiu5qQT7VN1kaPKx1hBV/e3HFzCen3XMVPerkOUEx0q2WrV0=
Authentication methods: password public_key
PMC> PMC> delete users slon password
password for user slon deleted
PMC>PMC> show users slon
Login: slon
Public key type: ecdsa-sha2-nistp256
Public key:
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEs+nzq8LrO0OeniYHT
YnTPtbc+CKaLAgvKz+3faiu5qQT7VN1kaPKx1hBV/e3HFzCen3XMVPerkOUEx0q2WrV0=
Authentication methods: public_key
PMC>
```

Deleting a user

To delete users, use the **delete** command:

```
PMC> delete users slon
User slon deleted
PMC> show users
------------------
User1:
Login: admin
Public key type:
Public key:
Authentication methods: password
------------------
PMC>
```

DASHBOARD

DashBoard

This section allows you to view the current state of DCFW, its load, the number of users, the amount of traffic passing through DCFW, the filtering systems operation, the license status, and so on. Reports are presented as widgets, which can be customized by the system administrator as required. You can add, delete, move, and resize widgets on the **Dashboard** page. By default, pages with NOC (Network Operation Center) and SOC (Security Operation Center) widgets already exist.

Some widgets allow you to customize the display, specify data filtering, and configure other settings. To configure a widget, click the gearwheel icon in the upper right corner. Not all parameters listed below are available for every type of widget.

Name	Description
Name	Name of widget to display in the Dashboard.
Description	Optional widget description.
Number of records	Maximum number of records to display.
Group by	Data field by which to group the data.
Chart	

Name	Description
	Select how the data is presented. Available values: • Number • Pie chart • Column chart • Bar chart • Table • Line chart • World map.
Filter query	SQL-like query string that allows you to limit the amount of information used to build a widget. To construct a query, use field names and values, keywords, and operators. For keywords and operators with examples of their use, see the Data Search and Filtering section.

Note

You can use highlighting to get a more detailed look at a specific part of the graph. To return you need to double-click with the left mouse button.

HELP

Help (Description)

This section provides links to useful resources in the UserGate support portal:

Name	Description
Help	Link to the latest version of the administrator's guide.
Help video	Link to a list of videos explaining how to configure various UserGate services.
Support	Link to the UserGate technical support portal at https://www.usergate.com/ru/support for more information about how to configure UserGate. This is also where you can submit a ticket to resolve your problem.

ADMIN

Amin (description)

This section allows registered administrators to change their passwords, update some profile settings and log out.

Name	Description
Change password	To change your password, enter your current password and then the new one twice.
Preferences	• Show items per page: number of lines to display in one dialog box, such as a list of firewall rules. • Night mode: set the dark theme for the UGOS GUI. • Favorite filters: rename or delete filters for various logs created by this user.
Logout	End the session in the web console of the device.

FAVORITES

Favorites

The web interface allows you to filter the displayed sections by adding them to favorites and search for sections by their name. You can use filtering to hide unused sections. Displaying only the favorite sections does not affect the device functionality or configuration. To add a section to favorites, click the asterisk next to the section name. To customize the display, use the **Favorites Only** switch at the bottom of the panel.

APPLICATIONS

Description of Log Formats

Logs Export in CEF Format

Event Log Format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log type.	events
	Origin	Module where the event occurred.	admin_console
	Severity	The severity of the event.	Available values: • 1: info • 4: warning • 7: error • 10: critical
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Event type.	login_successful

Field type	Field name	Description	Example value
	suser	The username.	Admin
	src	Source IPv4 address.	192.168.117.254
	cat	Component where the event occurred.	console_auth
	cs1Label	This field is used for event details.	Attributes
	cs1	Event details in JSON format.	{"name":"MIME_BUILTIN_COMPOSITE", "module":"nlist_import"}

Web access log format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	webaccess
	Name	Source type.	log
	Threat Level	Threat level for the URL category.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822

Field type	Field name	Description	Example value
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Action taken by the device according to the configured policies.	captive
	reason	The reason why the event was created, e.g. the reason for the site block.	{"id": 39,"name":"Social Networking","threat _level":3}
	proto	Level 4 protocol used.	TCP
	app	Application layer protocol and its version.	HTTP/1.1
	suser	The username.	user_example (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.10.10
	spt	Source port	Values: 0-65535.
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535.
	requestMethod	Method used to access the URL address (POST, GET, etc.).	GET
	request	In the case of an HTTP request, the field contains the URL of the requested	http:// www.secure.com

Field type	Field name	Description	Example value
		resource and the protocol used.	
	requestContext	Request source URL (HTTP referer).	https://www.google.com/
	requestClientApplication	Browser useragent.	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0
	in	Number of transmitted inbound bytes (data transferred from the source to the destination).	231
	out	Number of transmitted outbound bytes (data transferred from the destination to the source).	40
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	Default Allow
	cs2Label	Indicates the source zone.	Source Zone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the source country.	Source Country
	cs3	Source country name.	AE (a two-letter country code is displayed)
	cs4Label	Indicates the destination zone.	Destination Zone

Field type	Field name	Description	Example value
	cs4	Destination zone name.	Untrusted
	cs5Label	Indicates the destination country.	Destination Country
	cs5	Destination country name.	AE (a two-letter country code is displayed)
	cs6Label	Indicates if the content was decrypted.	Decrypted
	cs6	Decrypted or not.	true, false
	flexString1Label	Refers to the content type.	Media type
	flexString1	The type of the content.	text/html
	flexString2Label	Indicates the category of the requested URL.	URL Categories
	flexString2	URL category.	Computers & Technology
	cn1Label	Indicates the number of packets transmitted from the source to the destination.	Packets sent
	cn1	Number of packets transmitted from the source to the destination.	3
	cn2Label	Indicates the number of packets transmitted from the destination to the source.	Packets received
	cn2		1

Field type	Field name	Description	Example value
		Number of packets transmitted from the destination to the source.	
	cn3Label	Specifies the server's original response.	Response
	cn3	Status code.	302

CEF Compact Web Access Log Format:

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	webaccess
	Name	Source type.	log
	Threat Level	Threat level for the URL category.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Action taken by the device	captive

Field type	Field name	Description	Example value
		according to the configured policies.	
	reason	The reason why the event was created, e.g. the reason for the site block.	{"id": 39,"name":"Social Networking","threat_level":3}
	proto	Level 4 protocol used.	TCP
	app	Application layer protocol and its version.	HTTP/1.1
	suser	The username.	user_example (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.10.10
	spt	Source port	Values: 0-65535.
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535.
	requestMethod	Method used to access the URL address (POST, GET, etc.).	GET
	request	In the case of an HTTP request, the field contains the URL of the requested resource and the protocol used.	http://www.secure.com
	requestContext	Request source URL (HTTP referer).	https://www.google.com/

Field type	Field name	Description	Example value
	requestClientApplication	Browser useragent.	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0
	in	Number of transmitted inbound bytes (data transferred from the source to the destination).	231
	out	Number of transmitted outbound bytes (data transferred from the destination to the source).	40
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	Default Allow
	cs2Label	Indicates the source zone.	SrcZone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the destination zone.	DstZone
	cs3	Destination zone name.	Untrusted
	flexString1Label	Indicates the category of the requested URL.	URLCats
	flexString1	URL category.	Computers & Technology

Field type	Field name	Description	Example value
	cn1Label	Specifies the server's original response.	Response
	cn1	Status code.	302

> **Note**
> **Some field values are truncated to 80 characters, this is a general rule for the compact format. For example, a list of URL categories, URL, username, rule name, zone name, etc.**

DNS log format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	dns
	Name	Source type.	log
	Threat Level	Threat level for the URL category.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1701085036026
	deviceExternalId	The unique name of the device that generated the event.	utmcore@ntoorereaeda

Field type	Field name	Description	Example value
	act	Action taken by the device according to the configured policies.	block
	reason	The reason why the event was created, e.g. the URL category on which the rule was triggered.	{"url_cats":[{"id":37,"name":"Search Engines & Portals","threat_level":1}]}
	proto	Level 4 protocol used.	UDP
	dhost	The destination host name, whose address is determined using the DNS server.	google.com
	app	Application layer protocol	DNS
	suser	The username.	user1 (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.0.11
	spt	Source port	Values: 0-65535.
	smac	Source MAC address.	FA:16:3E:65:1C:B4
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535. Port 53 is normally used for DNS.
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1		Rule1

Field type	Field name	Description	Example value
		Name of the rule triggered to cause the event.	
	cs2Label	Indicates the source zone.	Source Zone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the source country.	Source Country
	cs3	Source country name.	AE (a two-letter country code is displayed)
	cs4Label	Indicates the destination zone.	Destination Zone
	cs4	Destination zone name.	Untrusted
	cs5Label	Indicates the destination country.	Destination Country
	cs5	Destination country name.	AE (a two-letter country code is displayed)
	cs6Label	Indicates the data being transmitted.	Data
	cs6	The transmitted data.	{"question": [{"domain":"google.com","type":"A","class":"IN"}], "answer": [{"domain":"google.com","type":"TXT","class": "IN","ttl": 5,"data":"Blocked"}, {"domain":"google.com", "type":"A","class":"IN","ttl":

Field type	Field name	Description	Example value
			5,"data":"10.10.0.1"}] }
	flexString1Label	Indicates the category of the requested URL.	URL Categories
	flexString1	URL category.	Search Engines & Portals

DNS log format **CEF Compact**:

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	dns
	Name	Source type.	log
	Threat Level	Threat level for the URL category.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1701085036026
	deviceExternalId	The unique name of the device that generated the event.	utmcore@ntoorereaeda
	act	Action taken by the device according to the	block

Field type	Field name	Description	Example value
		configured policies.	
	reason	The reason why the event was created, e.g. the URL category on which the rule was triggered.	{"url_cats":[{"id":37,"name":"Search Engines & Portals","threat_level":1}]}
	proto	Level 4 protocol used.	UDP
	dhost	The destination host name, whose address is determined using the DNS server.	google.com
	app	Application layer protocol	DNS
	suser	The username.	user1 (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.0.11
	spt	Source port	Values: 0-65535.
	smac	Source MAC address.	FA:16:3E:65:1C:B4
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535. Port 53 is normally used for DNS.
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	Rule1

Field type	Field name	Description	Example value
	cs2Label	Indicates the source zone.	SrcZone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the destination zone.	DstZone
	cs3	Destination zone name.	Untrusted
	cs4Label	Indicates the data being transmitted.	Data
	cs4	The transmitted data.	{"question":[{"domain":"google.com","type":"A","class":"IN"}], "answer":[{"domain":"google.com","type":"TXT","class":"IN","ttl":5,"data":"Blocked"},{"domain":"google.com","type":"A","class":"IN","ttl":5,"data":"10.10.0.1"}]}
	flexString1Label	Indicates the category of the requested URL.	URLCats
	flexString1	URL category.	Search Engines & Portals

Traffic log format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate

Field type	Field name	Description	Example value
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log type.	traffic
	Rule Type	Type of the rule triggered to cause the event.	firewall
	Threat Level	Application threat level.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Action taken by the device according to the configured policies.	accept
	proto	Level 4 protocol used.	TCP or UDP
	app	Triggered application name	my_app
	suser	The username.	user_example (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.10.10

Field type	Field name	Description	Example value
	spt	Source port	Values: 0-65535.
	smac	Source MAC address.	00:50:56:80:28:08
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535.
	dmac	Destination MAC address.	00:50:56:80:7D:21
	in	Number of transmitted inbound bytes (data transferred from the source to the destination).	231
	out	Number of transmitted outbound bytes (data transferred from the destination to the source).	40
	sourceTranslatedAddress	Source address after reassignment (if NAT rules are configured).	192.168.174.134 (0.0.0.0 if not)
	sourceTranslatedPort	Source port after reassignment (if NAT rules are configured).	Values: 0-65535 (0 if not)
	destinationTranslatedAddress	Destination address after reassignment (if NAT rules are configured).	192.226.127.130 (0.0.0.0 if not)
	destinationTranslatedPort	Destination port after reassignment (if NAT rules are configured).	Values: 0-65535 (0 if not)

Field type	Field name	Description	Example value
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	Allow trusted to untrusted
	cs2Label	Indicates the source zone.	Source Zone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the source country.	Source Country
	cs3	Source country name.	AE (a two-letter country code is displayed)
	cs4Label	Indicates the destination zone.	Destination Zone
	cs4	Destination zone name.	Untrusted
	cs5Label	Indicates the destination country.	Destination Country
	cs5	Destination country name.	AE (a two-letter country code is displayed)
	cn1Label	Indicates the number of packets transmitted from the source to the destination.	Packets sent
	cn1	Number of packets transmitted from the source to the destination.	3
	cn2Label	Indicates the number of packets transmitted from	Packets received

Field type	Field name	Description	Example value
		the destination to the source.	
	cn2	Number of packets transmitted from the destination to the source.	1

Traffic log format **CEF Compact**:

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log type.	traffic
	Rule Type	Type of the rule triggered to cause the event.	firewall
	Threat Level	Application threat level.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Action taken by the device according to the	accept

Field type	Field name	Description	Example value
		configured policies.	
	proto	Level 4 protocol used.	TCP or UDP
	app	Triggered application name	my_app
	suser	The username.	user_example (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.10.10
	spt	Source port	Values: 0-65535.
	smac	Source MAC address.	00:50:56:80:28:08
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535.
	dmac	Destination MAC address.	00:50:56:80:7D:21
	in	Number of transmitted inbound bytes (data transferred from the source to the destination).	231
	out	Number of transmitted outbound bytes (data transferred from the destination to the source).	40
	sourceTranslatedAddress	Source address after reassignment (if NAT rules are configured).	192.168.174.134 (0.0.0.0 if not)

Field type	Field name	Description	Example value
	sourceTranslatedPort	Source port after reassignment (if NAT rules are configured).	Values: 0-65535 (0 if not)
	destinationTranslatedAddress	Destination address after reassignment (if NAT rules are configured).	192.226.127.130 (0.0.0.0 if not)
	destinationTranslatedPort	Destination port after reassignment (if NAT rules are configured).	Values: 0-65535 (0 if not)
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	Allow trusted to untrusted
	cs2Label	Indicates the source zone.	SrcZone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the destination zone.	DstZone
	cs3	Destination zone name.	Untrusted

IDPS log format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7

Field type	Field name	Description	Example value
	Source	Log type.	idps
	Signature	Name of the triggered IPS signature.	BlackSun Test
	Threat Level	Signature threat level.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Action taken by the device according to the configured policies.	accept
	proto	Level 4 protocol used.	TCP or UDP
	app	Application layer protocol	HTTP
	suser	The username.	user_example (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.10.10
	spt	Source port	Values: 0-65535.
	dst	IPv4 address of the traffic destination.	194.226.127.130

Field type	Field name	Description	Example value
	dpt	Destination port	Values: 0-65535.
	in	Number of transmitted inbound bytes (data transferred from the source to the destination).	231
	out	Number of transmitted outbound bytes (data transferred from the destination to the source).	40
	msg	Signature threat level and name.	[2] BlackSun
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	IDPS Rule Example
	cs2Label	Indicates the source zone.	Source Zone
	cs2	Source zone name.	Trusted
	cs3Label	Indicates the source country.	Source Country
	cs3	Source country name.	AE (a two-letter country code is displayed)
	cs4Label	Indicates the destination zone.	Destination Zone
	cs4	Destination zone name.	Untrusted
	cs5Label	Indicates the destination country.	Destination Country

Field type	Field name	Description	Example value
	cs5	Destination country name.	AE (a two-letter country code is displayed)

IDPS log format **CEF Compact**:

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log type.	idps
	Signature	Name of the triggered IPS signature.	BlackSun Test
	Threat Level	Signature threat level.	Available values: 2, 4, 6, 8, 10 (the set threat level multiplied by 2); Unknown, if no category is defined.
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1652344423822
	deviceExternalId	The unique name of the device that generated the event.	utmcore@erstheta tica
	act	Action taken by the device according to the configured policies.	accept

Field type	Field name	Description	Example value
	proto	Level 4 protocol used.	TCP or UDP
	app	Application layer protocol	HTTP
	suser	The username.	user_example (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.10.10
	spt	Source port	Values: 0-65535.
	dst	IPv4 address of the traffic destination.	194.226.127.130
	dpt	Destination port	Values: 0-65535.
	in	Number of transmitted inbound bytes (data transferred from the source to the destination).	231
	out	Number of transmitted outbound bytes (data transferred from the destination to the source).	40
	msg	Signature threat level and name.	[2] BlackSun
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	IDPS Rule Example

Windows Active Directory Log Format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	endpoint_log
	Name	Source type.	log
	Threat Level	Threat level.	Available values: from 1 to 10 (the set threat level multiplied by 2).
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1701085036026
	deviceExternalId	The unique name of the device that generated the event.	utmcore@ntoorereaeda
	suser	The username.	user1.dep.local
	msg	The event description in the AD log.	Group membership information Subject: Security ID: S-1-0-0 Account Name: — Account Domain: — Logon ID: 0x0 Logon Type: 3 New Logon: Security ID: S-1-5-21-3795870133-5220325-2125745684-1103 Account Name: user1 Account Domain:

Field type	Field name	Description	Example value
			DEP Logon ID: 0xA57A446 Event in sequence: 1 of 1 Group Membership: % {S-1-5-21-3795870133-5220325-2125745684-513} % {S-1-1-0} % {S-1-5-32-544} % {S-1-5-32-555} % {S-1-5-32-545} % {S-1-5-32-554} % {S-1-5-2} % {S-1-5-11} % {S-1-5-15} % {S-1-5-21-3795870133-5220325-2125745684-512} % {S-1-5-21-3795870133-5220325-2125745684-572} % {S-1-5-64-10} % {S-1-16-12288} The subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe. The logon type field indicates the kind of logon that occurred. The most common types are 2 (interactive) and 3 (network). The New Logon fields indicate the account for whom the new logon was created, i.e. the account that was logged on. This

Field type	Field name	Description	Example value
			event is generated when the Audit Group Membership subcategory is configured. The Logon ID field can be used to correlate this event with the corresponding user logon event as well as to any other security audit events generated during this logon session.
	cn1Label	Indicates the event code in the AD log.	logEventCode
	cn1	Event code.	4627
	cn2Label	Indicates the event ID in the AD log.	logEventId
	cn2	Event ID.	4627
	cn3Label	Indicates the event type in the Windows log (System\Security\ Application, etc.).	logEventType
	cn3	Windows log event type.	4
	cs1Label	Indicates the ID of the endpoint — the source of the event.	endpointId
	cs1	The endpoint ID.	16535060-5a1a-4e92-8331-239406ec34da
	cs2Label	Indicates the name of the endpoint — the source of the event (UserGate	endpointName

Field type	Field name	Description	Example value
		client, WMI sensor, etc.).	
	cs2	Endpoint device name.	dep.local
	cs3Label	Indicates the severity of the event in the AD log.	logLevel
	cs3	Event severity level.	Audit Success
	cs4Label	Indicates the event category code (12554 Group Membership, 12544 Logon, 14337 Kerberos Service Ticket Operations, etc.).	logCategoryString
	cs4	The event's category.	Group Membership
	cs5Label	Indicates the Windows log file.	logFile
	cs5	Windows log file	Security
	cs6Label	Indicates the source of the AD log.	sourceName
	cs6	The source of the AD log.	Microsoft-Windows-Security-Auditing
	flexString1Label	Indicates the content of the event in the AD log.	insertionString
	flexString1	Parameters of the AD log event after message parsing.	['S-1-0-0', '-', '-', '0x0', 'S-1-5-21-3795870133-5220325-2125745684-1103', 'user1', 'DEP', '0x7a25a21',

Field type	Field name	Description	Example value
			'3', '1', '1', '\\r\\n\\t\ \t% {S-1-5-21-3795870133-5220325-2125745684-513}\\r\\n\ \t\\t%{S-1-1-0}\\r\ \n\\t\\t% {S-1-5-32-544}\\r\ \n\\t\\t% {S-1-5-32-555}\\r\ \n\\t\\t% {S-1-5-32-545}\\r\ \n\\t\\t% {S-1-5-32-554}\\r\ \n\\t\\t%{S-1-5-2}\ \r\\n\\t\\t% {S-1-5-11} \\r\\n\\t\\t% {S-1-5-15}\\r\\n\\t\ \t% {S-1-5-21-3795870133-5220325-2125745684-512}\\r\\n\ \t\\t% {S-1-5-21-3795870133-5220325-2125745684-572}\\r\\n\ \t\\t% {S-1-5-64-10}\\r\ \n\\t\\t% {S-1-16-12288}']

Syslog Format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	syslog
	Name	Source type.	log

Field type	Field name	Description	Example value
	Threat Level	Threat level.	Available values: • 0: emergencies • 1: alerts • 2: critical • 3: errors • 4: warnings • 5 — notifications; • 6 — informational; • 7: debugging
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1701085036026
	deviceExternalId	The unique name of the device that generated the event.	utmcore@ntoorereaeda
	msg	The event description.	[3603:3603:1128/175000.938565:ERROR:CONSOLE(6)] "console.assert", source: devtools://devtools/bundled/devtools-frontend/front_end/panels/console/console.js (6)
	cn1Label	Indicates the source type of Syslog events. For more information about Syslog facility values, see RFC 5424.	Facility

Field type	Field name	Description	Example value
	cn1	Syslog event source type. Example: user-level messages.	1
	cs1Label	Indicates the name of the device where the event occurred.	Hostname
	cs1	The name of the computer where the event occurred.	node1
	cs2Label	Indicates the application that caused the event.	Tag
	cs2	The application that caused the event.	org.gnome.Shell.desktop
	cs3Label	Indicates the process ID of the event.	ProcessID
	cs3	PID of the process triggering the event.	3036
	cs4Label	Indicates that a rule was triggered.	Rule
	cs4	Name of the rule triggered to cause the event.	Example: Allow user-level messages

RADIUS log format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW

Field type	Field name	Description	Example value
	Device Version	Product version.	7
	Source	Log name.	radius
	Name	Source type.	log
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1701085036026
	deviceExternalId	The unique name of the device that generated the event.	utmcore@ntoorereaeda
	act	User status (acct_status_type).	start, stop, interim update, accounting-on, accounting-off
	suser	The username.	Unknown, if the user is unknown.
	src_ip	The IP address of the source where the message came from.	192.168.57.4
	dst	The IP address of the NAS that authorized the user.	172.16.1.4
	dvc	User IP address (framed IP address).	192.168.57.29
	cs1Label	Indicates the group the user is a member of.	user groups
	cs1	A string of groups the user is a member of.	test_group

UserID log format

Field type	Field name	Description	Example value
CEF header	**CEF:Version**	CEF version.	CEF:0
	Device Vendor	Product vendor.	UserGate
	Device Product	Product type.	NGFW
	Device Version	Product version.	7
	Source	Log name.	userid
	Name	Source type.	log
CEF [extension]	**rt**	Time when the event was received (in milliseconds since January 1, 1970).	1701085036026
	deviceExternalId	The unique name of the device that generated the event.	utmcore@ntoorereaeda
	act	Action taken by the device according to the configured policies.	login
	reason	The reason why the event was created.	{"user_groups_sids": ["S-1-5-21-3795870133-5220325-2125745684-513","S-1-5-21-3795870133-5220325-2125745684-512"], "user_sid":"S-1-5-21-3795870133-5220325-2125745684-1103","login":"user1","domain":"DEV","event_id":4624}
	suser	The username.	

Field type	Field name	Description	Example value
			user1 (Unknown, if the user is unknown)
	src	Traffic source IPv4 address.	10.10.0.11
	cs1Label	Indicates that a rule was triggered.	Rule
	cs1	Name of the rule triggered to cause the event.	dev.local

Export logs in JSON format

Event log description

Field name	Description	Example value
timestamp	Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
node	The unique name of the device that generated the event.	utmcore@ersthetatica
ip_address	IPv4 address of the event source.	192.168.174.134
attributes	Event details in JSON format.	{"rule":{"logrotate": 12,"attributes": {"timezone":"UAE/ Dubai"},"id":"66f9de9f-d698-4bec-b3b0-ba65b46d3608","name":"Example log export ftp"}
event_type	Event type.	logexport_rule_updated
event_severity	Event severity.	info (informational), warning (warnings), error (errors), critical (critical).

Field name	Description	Example value
event_origin	A module where the event occurred.	core
event_component	A component where the event occurred.	console_auth
user	Username.	{"guid":"37333739-3733-3734-3635-366400000000","name":"System","groups":[]}}

Web access log description

Field name	Description	Example value
timestamp	Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
session	Session ID.	a7a3cd49-8232-4f1a-962a-3659af89e96f (if System: 00000000-0000-0000-0000-000000000000)
node	The unique name of the device that generated the event.	utmcore@ersthetatica
reasons	The reason why the event was created, e.g. the reason for the site block.	"url_cats":[{"id":39,"name":"Social Networking","threat_level":3}]
proto	Level 4 protocol used.	TCP
host	Hostname.	www.google.com
action	Action taken by the device according to the configured policies.	block
bytes_sent	Number of bytes transmitted from the source to the destination.	52
bytes_recv	Number of packets transmitted from the destination to the source.	100

Field name		Description	Example value
packets_sent		Number of packets transmitted from the source to the destination.	2
packets_recv		Number of bytes transmitted from the destination to the source.	5
request_method		Method used to access the URL address (POST, GET, etc.).	GET
url		Contains the URL of the requested resource and the protocol used.	http://www.secure.com
media_type		The type of the content.	application/json
status_code		Status code.	302
http_referer		Request source URL (HTTP referer).	https://www.google.com/
decrypted		Indicates if the content was decrypted.	true, false
useragent		Browser useragent.	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/ 20100101 Firefox/96.0
application	**id**	Application ID.	20
	name	Application name.	Youtube
	threat_ level	Application threat level.	0
	app_pr otocol	Application layer protocol and its version.	HTTP\/1.1"
url_cat egories	**id**	ID of the category to which the URL belongs.	39
	threat_level	Threat level for the URL category.	Available values: • 1: very low • 2: low • 3: medium

Field name			Description	Example value
				• 4: high • 5: very high
	name		Name of the category to which the URL belongs.	Social Networking
source	**zone**	**guid**	Unique ID of the traffic source zone.	d0038912-0d8a-4583-a525-e63950b1da47
		name	Source zone name.	Trusted
	country		Traffic source country.	AE (a two-letter country code is displayed)
	ip		Source IPv4 address.	10.10.10.10
	port		Source port	Values: 0-65535.
	mac		source MAC address	01:23:45:67:89:AB
destination	**zone**	**guid**	Unique ID of the traffic destination zone.	3c0b1253-f069-4060-903b-5fec4f465db0
		name	Traffic destination zone name.	Untrusted
	country		The destination country.	AE (a two-letter country code is displayed)
	ip		Destination IPv4 address.	192.168.174.134
	port		Destination port	Values: 0-65535.
	mac		Destination MAC address.	01:23:45:67:89:AB
rule	**guid**		Unique ID of the rule triggered to cause the event.	f93da24d-74f9-4f8c-9e9b-8e6d02346fb4
	name		The name of the rule.	Default allow
	type		Triggered rule type.	
user	**guid**		Unique ID of the user.	a7a3cd49-8232-4f1a-962a-3659af89e96f
	name		Username.	user_name

Field name			Description	Example value
	groups	**guid**	Unique ID of the group the user is a member of.	919878b2-e882-49ed-3331-8ec72c3c79cb
		name	Name of the group the user is a member of.	Default Group

DNS log description

Field name		Description	Example value
timestamp		Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
session		Session ID.	00000000-0000-0000-0000-000000000000
node		The unique name of the device that generated the event.	utmcore@ntoorereaeda
reasons		The reason why the event was created, e.g. the URL category on which the rule was triggered.	{"url_cats":[{"id": 37,"name":"Search Engines & Portals","threat_level":1}]}
proto		Level 4 protocol used.	UDP
host		Hostname.	google.com
data		Indicates the data being transmitted.	{"question": [{"domain":"google.com","type":"A","class":"IN"}], "answer": [{"domain":"google.com","type":"TXT","class": "IN","ttl":5,"data":"Blocked"}, {"domain":"google.com", "type":"A","class":"IN","ttl": 5,"data":"10.10.0.1"}]}
url_categories	**id**	ID of the triggered URL category.	37

Field name			Description	Example value
	threat_level		Threat level of the triggered category.	Available values: • 1: very low • 2: low • 3: medium • 4: high • 5: very high
	name		Name of the triggered category.	Search Engines & Portals
action			Action taken by the device according to the configured policies.	block
application		**id**	Application ID.	5
		name	Application name.	
		threat_level	Application threat level.	0
		app_protocol	Application layer protocol	DNS
source	**zone**	**guid**	Unique ID of the traffic source zone.	d0038912-0d8a-4583-a525-e63950b1da47
		name	Traffic source zone name.	Trusted
	country		Source country name.	AE (a two-letter country code is displayed)
	ip		IPv4 address of the traffic source.	10.10.10.10
	port		Source port	Values: 0-65535.
	mac		Source MAC address.	01:23:45:67:89:AB
destination	**zone**	**guid**	Unique ID of the traffic destination zone.	3c0b1253-f069-4060-903b-5fec4f465db0
		name	Traffic destination zone name.	Untrusted

Field name			Description	Example value
	country		Destination country name.	AE (a two-letter country code is displayed)
	ip		IPv4 address of the traffic destination.	104.19.197.151
	port		Destination port.	Values: 0-65535. Port 53 is normally used for DNS.
	mac		Destination MAC address	01:23:45:67:89:AB
rule	**guid**		Unique ID of the rule triggered to cause the event.	59e38e06-533a-4771-9664-031c3e8b2e1f
	name		Name of the rule triggered to cause the event.	Rule1
	Type		Triggered rule type.	
user	**guid**		Unique ID of the user. If the user type is Unknown then the ID: 00000000-0000-0000-0000-000000000000.	a7a3cd49-8232-4f1a-962a-3659af89e96f
	name		The username.	user1
	groups	**guid**	Unique ID of the group the user is a member of.	919878b2-e882-49ed-3331-8ec72c3c79cb
		name	Name of the group the user is a member of.	Default Group

Traffic log description

Field name	Description	Example value
timestamp	Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
session	Session ID.	a7a3cd49-8232-4f1a-962a-3659af89e96f (if System: 00000000-0000-0000-0000-000000000000)

Field name			Description	Example value
node			The unique name of the device that generated the event.	utmcore@ersthetatica
proto			Level 4 protocol used.	TCP or UDP
action			Action taken by the device according to the configured policies.	accept
bytes_sent			Number of bytes transmitted from the source to the destination.	100
bytes_recv			Number of bytes transmitted from the destination to the source.	6
packets_recv			Number of packets transmitted from the destination to the source.	1
packets_sent			Number of packets transmitted from the source to the destination.	1
json_data			Additional data.	null
application	**id**		Application ID.	195
	threat_level		Application threat level.	Available values: • 1: very low • 2: low • 3: medium • 4: high • 5: very high
	app_protocol		Application layer protocol	HTTP
	name		Application name.	Youtube
source	**zone**	**guid**	Unique ID of the traffic source zone.	d0038912-0d8a-4583-a525-e63950b1da47
		name	Traffic source zone name.	Trusted

Field name			Description	Example value
	country		Source country name.	AE (a two-letter country code is displayed)
	ip		IPv4 address of the traffic source.	10.10.10.10
	port		Source port	Values: 0-65535.
destination	**zone**	**guid**	Unique ID of the traffic destination zone.	3c0b1253-f069-4060-903b-5fec4f465db0
		name	Traffic destination zone name.	Untrusted
	country		Destination country name.	AE (a two-letter country code is displayed)
	ip		IPv4 address of the traffic destination.	104.19.197.151
	port		Destination port.	Values: 0-65535.
nat	**source**	**ip**	Source address after reassignment (if NAT rules are configured).	192.168.117.85 (if NAT is not configured then **"nat":null**)
		port	Source port after reassignment (if NAT rules are configured).	Values: 0-65535 (if NAT is not configured then **"nat":null**)
	destination	**ip**	Destination address after reassignment (if NAT rules are configured).	64.233.164.198 (if NAT is not configured then **"nat":null**)
		port	Source port after reassignment (if NAT rules are configured).	Values: 0-65535 (if NAT is not configured then **"nat":null**)
rule	**guid**		Unique ID of the rule triggered to cause the event.	59e38e06-533a-4771-9664-031c3e8b2e1f
	type		Rule type.	firewall
	name		Name of the rule triggered to cause the event.	Allow trusted to untrusted

Field name			Description	Example value
user	**guid**		Unique ID of the user. If the user type is Unknown then the ID: 00000000-0000-0000-0000-000000000000.	a7a3cd49-8232-4f1a-962a-3659af89e96f
	name		The username.	Admin
	groups	**guid**	Unique ID of the group the user is a member of.	919878b2-e882-49ed-3331-8ec72c3c79cb
		name	Name of the group the user is a member of.	Default Group

IDPS log description

Field name	Description	Example value
timestamp	Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
session	Session ID.	a7a3cd49-8232-4f1a-962a-3659af89e96f (if System: 00000000-0000-0000-0000-000000000000)
node	The unique name of the device that generated the event.	utmcore@ersthetatica
proto	Level 4 protocol used.	TCP or UDP
action	Action taken by the device according to the configured policies.	accept
bytes_sent	Number of bytes transmitted from the source to the destination.	100
bytes_recv	Number of bytes transmitted from the destination to the source.	6
packets_sent		1

Field name			Description	Example value
			Number of packets transmitted from the source to the destination.	
packets_recv			Number of packets transmitted from the destination to the source.	1
json_data			Additional data.	null
application	**id**		Application ID.	195
	threat_level		Application threat level.	Available values: • 1: very low • 2: low • 3: medium • 4: high • 5: very high
	name		Application name.	Youtube
	app_protocol		Application layer protocol	HTTP
user	**guid**		Unique ID of the user. If the user type is Unknown then the ID: 00000000-0000-0000-0000-000000000000.	a7a3cd49-8232-4f1a-962a-3659af89e96f
	name		The username.	Admin
	groups	**guid**	Unique ID of the group the user is a member of.	919878b2-e882-49ed-3331-8ec72c3c79cb
		name	Name of the group the user is a member of.	Default Group
rule	**guid**		Unique ID of the rule triggered to cause the event.	59e38e06-533a-4771-9664-031c3e8b2e1f
	name		Name of the rule triggered to cause the event.	Allow trusted to untrusted
	type		Triggered rule type	idps

Field name			Description	Example value
signatures	**id**		ID of the triggered signature.	999999
	threat_level		Threat level of the triggered signature.	Available values: • 1: very low • 2: low • 3: medium • 4: high • 5: very high
	name		Name of the triggered signature.	BlackSun Test
source	**zone**	**guid**	Unique ID of the traffic source zone.	d0038912-0d8a-4583-a525-e63950b1da47
		name	Traffic source zone name.	Trusted
	country		Source country name.	AE (a two-letter country code is displayed)
	ip		IPv4 address of the traffic source.	10.10.10.10
	port		Source port	Values: 0-65535.
	mac		Source MAC address.	01:23:45:67:89:AB
destination	**zone**	**guid**	Unique ID of the traffic destination zone.	3c0b1253-f069-4060-903b-5fec4f465db0
		name	Traffic destination zone name.	Untrusted
	country		Destination country name.	AE (a two-letter country code is displayed)
	ip		IPv4 address of the traffic destination.	104.19.197.151
	port		Destination port	Values: 0-65535.
	mac		Destination MAC address.	01:23:45:67:89:AB

Windows Active Directory Log Description

Field name	Description	Example value
timestamp	Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
node_name	A name that uniquely identifies the UserGate device generating this event.	utmcore@ntoorereaeda
endpoint_id	ID of the endpoint that is the source of the event.	16535060-5a1a-4e92-8331-239406ec34da
endpoint_name	Name of the endpoint that is the source of the event.	dep.local
user_name	The "User" field from AD log.	user1.dep.local
log_level	The "Keywords" field from AD log.	Audit Success
log_category_string	Event category code in the AD log.	Group Membership
log_file	Windows log file.	Security
source_name	The "Source" field from AD log.	Microsoft-Windows-Security-Auditing
data	Event description in the AD log.	Group membership information.\r\n\r\nSubject: \r\n\tSecurity ID: \t\tS-1-0-0\r\n\tAccount Name:\t\t-\r\n\tAccount Domain:\t\t-\r\n\tLogon ID: \t\t0x0\r\n\r\nLogon Type: \t\t\t3\r\n\r\nNew Logon: \r\n\tSecurity ID: \t\tS-1-5-21-3795870133-5220325-2125745684-1103\r\n\tAccount Name: \t\tuser1\r\n\tAccount Domain:\t\tDEP\r\n\tLogon ID: \t\t0x7A25A21\r\n\r\nEvent in sequence:\t\t1 of 1\r\n\r\nGroup Membership: \t\t\t\r\n\t\t% {S-1-5-21-3795870133-522032

Field name	Description	Example value
		5-2125745684-513}\r\n\t\t% {S-1-1-0}\r\n\t\t% {S-1-5-32-544}\r\n\t\t% {S-1-5-32-555}\r\n\t\t% {S-1-5-32-545}\r\n\t\t% {S-1-5-32-554}\r\n\t\t% {S-1-5-2}\r\n\t\t% {S-1-5-11}\r\n\t\t% {S-1-5-15}\r\n\t\t% {S-1-5-21-3795870133-5220325-2125745684-512}\r\n\t\t% {S-1-5-21-3795870133-5220325-2125745684-572}\r\n\t\t% {S-1-5-64-10}\r\n\t\t% {S-1-16-12288}\r\n\r\nThe subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe.\r\n\r\nThe logon type field indicates the kind of logon that occurred. The most common types are 2 (interactive) and 3 (network).\r\n\r\nThe New Logon fields indicate the account for whom the new logon was created, i.e. the account that was logged on.\r\n\r\nThis event is generated when the Audit Group Membership subcategory is configured. The Logon ID field can be used to correlate this event with the corresponding user logon event as well as to any other security audit events generated during this logon session.
computer_name	Windows node from the AD log where the event took place.	DC1.dep.local
insertion_string	Parameters of the AD log event after message parsing.	['S-1-0-0', '-', '-', '0x0', 'S-1-5-21-3795870133-5220325-2125745684-1103', 'user1',

Field name	Description	Example value
		'DEP', '0x7a25a21', '3', '1', '1', '\\r\\n\\t\\t% {S-1-5-21-3795870133-5220325-2125745684-513}\\r\\n\\t\\t%{S-1-1-0}\\r\\n\\t\\t% {S-1-5-32-544}\\r\\n\\t\\t% {S-1-5-32-555}\\r\\n\\t\\t% {S-1-5-32-545}\\r\\n\\t\\t% {S-1-5-32-554}\\r\\n\\t\\t% {S-1-5-2}\\r\\n\\t\\t%{S-1-5-11} \\r\\n\\t\\t%{S-1-5-15}\\r\\n\\t\\t% {S-1-5-21-3795870133-5220325-2125745684-512}\\r\\n\\t\\t% {S-1-5-21-3795870133-5220325-2125745684-572}\\r\\n\\t\\t%{S-1-5-64-10}\\r\\n\\t\\t% {S-1-16-12288}']
error	Error code from the AD log that occurred while receiving data.	0
status	Error description from the AD log that occurred while receiving data.	
counter_id	Counter ID of the WMI sensor.	login_logout
log_event_code	The "Event code" field from AD log.	4627
log_event_id	The "Event ID" field from AD log.	4627
log_event_type	Windows log even type (System/Security/Application etc.)	4

Syslog Description

Field name	Description	Example value
timestamp	Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z

Field name		Description	Example value
node		The unique name of the device that generated the event.	utmcore@ntoorereaeda
syslog_facility		Syslog event source type. Example: user-level messages. For more information about Syslog facility values, see RFC 5424.	1
syslog_severity		Syslog event severity level. Example: warning. For more information about Syslog severity values, see RFC 5424.	4
computer_name		The name of the device where the event occurred.	node1
app_name		Application triggering the event.	org.gnome.Shell.desktop
process_id		PID of the process triggering the event.	3036
data		The event description.	[3603:3603:1130/125201.838651:ERROR:CONSOLE(6)] \"console.assert\", source: devtools://devtools/bundled/devtools-frontend/front_end/panels/console/console.js (6)
rule	**guid**	Unique ID of the rule triggered to cause the event.	16535060-5a1a-4e92-8331-239406ec34da
	name	Name of the rule triggered to cause the event.	Example: Allow user-level messages
	type	Triggered rule type.	

RADIUS log description

Field name		Description	Example value
timestamp		Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
node		The unique name of the device that generated the event.	utmcore@ntoorereaeda
event_type		User status (acct_status_type).	start, stop, interim update, accounting-on, accounting-off
action		Action taken by the device according to the configured policies.	login
src_ip		The IP address of the source where the message came from.	192.168.57.4
nas_ip		The IP address of the NAS that authorized the user.	172.16.1.4
framed_ip		User's IP address.	192.168.57.29
rule	**guid**	Unique ID of the rule triggered to cause the event.	16535060-5a1a-4e92-8331-239406ec34da
user	**guid**	Unique ID of the user. If the user type is Unknown then the ID: 00000000-0000-0000-0000-000000000000.	745591c3-9d21-092d-8db4-5b9b0000044f
	name	The username.	user_name
	groups	Name of the group the user is a member of.	test_group

UserID log description

Field name		Description	Example value
timestamp		Time when the event was received. Format: yyyy-mm-ddThh:mm:ssZ.	2022-05-12T08:11:46.15869Z
node		The unique name of the device that generated the event.	utmcore@ntoorereaeda
reasons		The reason why the event was created.	{\"user_groups_sids\": [\"S-1-5-21-3795870133-522032 5-2125745684-513\", \"S-1-5-21-3795870133-522032 5-2125745684-512\", \"S-1-5-21-3795870133-522032 5-2125745684-572\"], \"user_sid\": \"S-1-5-21-3795870133-522032 5-2125745684-1103\",\"login\": \"user1\",\"domain\":\"DEV\", \"event_id\":4624}
action		Action taken by the device according to the configured policies.	login
src_ip		IPv4 address of the event source.	10.10.0.11
rule	**guid**	Unique ID of the rule triggered to cause the event.	16535060-5a1a-4e92-8331-23 9406ec34da
	name	Name of the rule triggered to cause the event.	dev.local
	type	Triggered rule type.	syslog
user	**guid**	Unique ID of the user. If the user type is Unknown then the ID: 00000000-0000-0000-000 0-000000000000.	745591c3-9d21-092d-8db4-5 b9b0000044f
	name	The username.	user1

Field name			Description	Example value
	groups	**guid**	Unique ID of the group the user is a member of.	aa218609-8716-9252-df20-88c43a0d0bf6
		name	Name of the group the user is a member of.	CN=Domain Users,CN=Users,DC=dev,DC=local

Network Environment Requirements

Service	Protocol	Port	Outbound/ Inbound	Function
Web console	TCP	8001	Inbound (to UserGate DCFW Web Console)	Access to the management web interface of a device.
CLI over SSH	TCP	2200	Inbound (to CLI over SSH)	Access to the UserGate command line interface (CLI) over SSH.
XML-RPC	TCP	4040	Inbound (to UserGate via API)	UserGate device management via API.

Service	Protocol	Port	Outbound/ Inbound	Function
Remote assistance	TCP	22	Outbound (to technical support servers)	Remote access to technical support servers. Access to servers: • 93.91.171.46; • 178.154.221.222; • ra.entensys.com.
NTP	UDP	123	Outbound (to a precision time server)/ Inbound (from clients to the UserGate server, if it is used as a precision time server)	Time synchronization.
DNS	TCP/UDP	53	Inbound (from clients to the UserGate server, if it is acting as a DNS server)	The service that resolves domain names into IP addresses.
	UDP	53	Outbound (to DNS servers)	
UserGate server registration	TCP	443	Outbound (to the registration server)	UserGate product registration: access to reg2.usergate.com.

Service	Protocol	Port	Outbound/ Inbound	Function
Update software and libraries	TCP	443	Outbound (to update servers)	Update software and library items: access to updates.usergate.com.
Replicate settings	TCP	4369	Inbound (from the first cluster node to the second and subsequent nodes)	This service is required for the configuration cluster to work. Set up a control connection.
		9000-9100	Inbound (receive configuration from the first cluster node)	Transmit information about cluster configuration changes (replicate settings).
Communication with UserGate Management Center	TCP	9712	Outbound (from UG DCFW to UGMC)	Initial communication and encryption key exchange with the UserGate Management Center server.
		2022	Outbound (from UG DCFW to UGMC)	Build an SSH tunnel to exchange data using the received keys.
Communication with UserGate Log Analyzer	TCP	9713	Inbound (from LogAn to UG DCFW)	Initial communication and exchange of encryption

Service	Protocol	Port	Outbound/ Inbound	Function
				keys with the UserGate Log Analyzer server.
		2023	Inbound (from LogAn to UG DCFW)	Build an SSH tunnel to exchange data using the received keys.
	TCP	For versions 6.1.x: 1269 (transmit data to LogAn 6.1.x), 22699 (transmit data to LogAn 7.x.x) For versions 7.0.x: 22699 (transmit data to LogAn 6.1.x), 22711 (transmit data to LogAn 7.x.x using SSL)	Outbound (from UG DCFW to LogAn)	Transmit logs and telemetry to LogAn server.
Connection of endpoints with UserGate Client software installed (available starting from version 7.1.0)	TCP	4045	Inbound (from an endpoint device to UG DCFW)	Connecting endpoints and receiving telemetry to check compliance.
LDAP	TCP	389, 636	Outbound (to LDAP connector)	Execute LDAP requests (389 for LDAP and 636 for LDAP over SSL).
Captive portal and block pages	TCP	80, 443, 8002	Inbound (from a client	Display a Captive portal

Service	Protocol	Port	Outbound/ Inbound	Function
			browser to UG DCFW)	authenticatio n page and block pages.
		8043		When the "HTTPS for auth page" option is activated.
Kerberos	TCP/UDP	88	Outbound (to a Kerberos authenticatio n server)	Authenticate users via the Kerberos protocol.
NTLM	TCP	445	Outbound (to an NTLM authenticatio n server)	Authenticate users via the NTLM protocol.
RADIUS	UDP	1812	Outbound (to a RADIUS authenticatio n server)	User authenticatio n via the RADIUS protocol.
TACACS+	TCP	49	Outbound (to a TACACS+ authenticatio n server)	User authenticatio n via the TACACS+ protocol.
Terminal service agent	UDP	1812, 1813	Inbound (from the agent to UG DCFW)	Access to the UserGate server required for the terminal agent to work.
Windows Authentication Agent	UDP	1812, 1813	Inbound (from the agent to UG DCFW)	Access to the UserGate server required for the authenticatio n agent to work for Windows OS

Service	Protocol	Port	Outbound/ Inbound	Function
				domain users.
Proxy agent	UDP	8090	Inbound (from the agent to UG DCFW)	Access to the UserGate server required for the proxy agent to provide Internet access to Windows OS users.
SNMP	UDP	161	Inbound (to UserGate)	Access to the UserGate server via SNMP.
SMTP	TCP	25	Outbound (to the mail server)	Send alerts to email.
ICAP	TCP	1344	Outbound (to ICAP servers)	Service to work with ICAP servers.
DHCP	UDP	67, 68	Outbound (requesting an address from UserGate to a DHCP server)/ Inbound (UserGate acts as a DHCP server)	DHCP service.
BGP	TCP	179	Outbound (send information to neighbor BGP routers)/ Inbound (receive information from	BGP dynamic routing service.

Service	Protocol	Port	Outbound/ Inbound	Function
			neighbor BGP routers)	
OSPF	89/OSPF		Outbound (send information to neighbor OSPF routers)/ Inbound (receive information from neighbor OSPF routers)	OSPF dynamic routing service.
RIP	UDP	520	Outbound (distribute RIP routes to neighbor routers)/ Inbound (receive RIP routes from neighbor routers)	RIP dynamic routing service.
FTP (logs export)	TCP	21	Outbound (to an FTP server)	Export logs to an FTP server.
SSH (logs export)	TCP	22	Outbound (to an SSH server)	Export logs to an SSH server.
Syslog (logs export)	TCP/UDP	514	Outbound (to the Syslog server)	Export logs to a Syslog server.

DHCP options

Option value format corresponds to RFC 2132.

Name	Description
1	Subnet mask for the subnet from which the address was received.
2	Time difference between the time of the client subnet and UTC time (specified in seconds).
3	List of IP addresses of available gateways.
6	DNS server list.
7	Log server list (MIT-LCS UDP).
9	LPR server list (RFC 1179).
13	Boot image size for clients.
15	Domain name
16	Swap server.
17	The path to the client root directory.
18	The path to BOOTP extensions.
19	IP datagram forwarding status.
20	Remote source routing status.
21	IP address filtration policy.
22	Datagram maximum size.
23	Default TTL value for IP protocol.
26	MTU value for the given interface.
27	An option showing that all subnets use current MTU configuration.
31	An option which defines using ICMP messages for router detection.
32	Address which is used to access the router.
33	Routing static list; contains pairs "destination address" — "router address".

Name	Description
34	Using trailers for ARP requests.
35	Timeout for ARP cache memory.
36	An option showing whether it is necessary to use Ethernet data encapsulation.
37	TTL value for TCP packets.
38	Interval for sending TCP control packets (TCP keep-alive).
40	NIS domain.
41	NIS server list.
42	NTP time server list.
44	List of IP addresses of NetBIOS servers.
45	List of IP addresses of NetBIOS datagram forwarding servers.
46	NetBIOS node type.
47	NetBIOS area.
48	IP addresses of X Windows font servers (X Window System Font).
49	X Windows display manager.
58	T1 — the time interval during which the client should send a request to update the IP address.
59	T2 — the time interval (in seconds) during which the client should send a request for rebinding.
60	This option is used by DHCP client to set a vendor.
64	NIS+ domain name.
65	NIS+ server list.
66	TFTP server name.
67	Boot file name.

Name	Description
68	Addresses of home agents (Mobile IP Home Agent).
69	SMTP server list.
70	POP3 server list.
71	NNTP server list.
74	IRC server list.
77	User class.
80	This option allows to receive the network settings from the DHCP server by the fast exchange of two messages instead of the usual four messages between the Requesting Router (RR) and the Delegating Router (DR).
93	DHCP client system architecture.
94	DHCP client network interface ID.
97	Client ID based on UUID/GUID.
119	DNS lookup list.
120	SIP server list.
121	The list of classless static routes.
125	This option is used to specify information about a vendor.
255	The end of the option list; must be specified last.

Installing Local CA Certificates

Download the authorization authority certificate that you use to intercept HTTPS traffic, as described in the Certificate Management chapter, and follow the instructions for installing the certificate later in this section.

Installing a Certificate in Internet Explorer, Chrome Browsers in Windows OS

Open the folder where you downloaded the pem certificate, rename it user.der and double click on it:

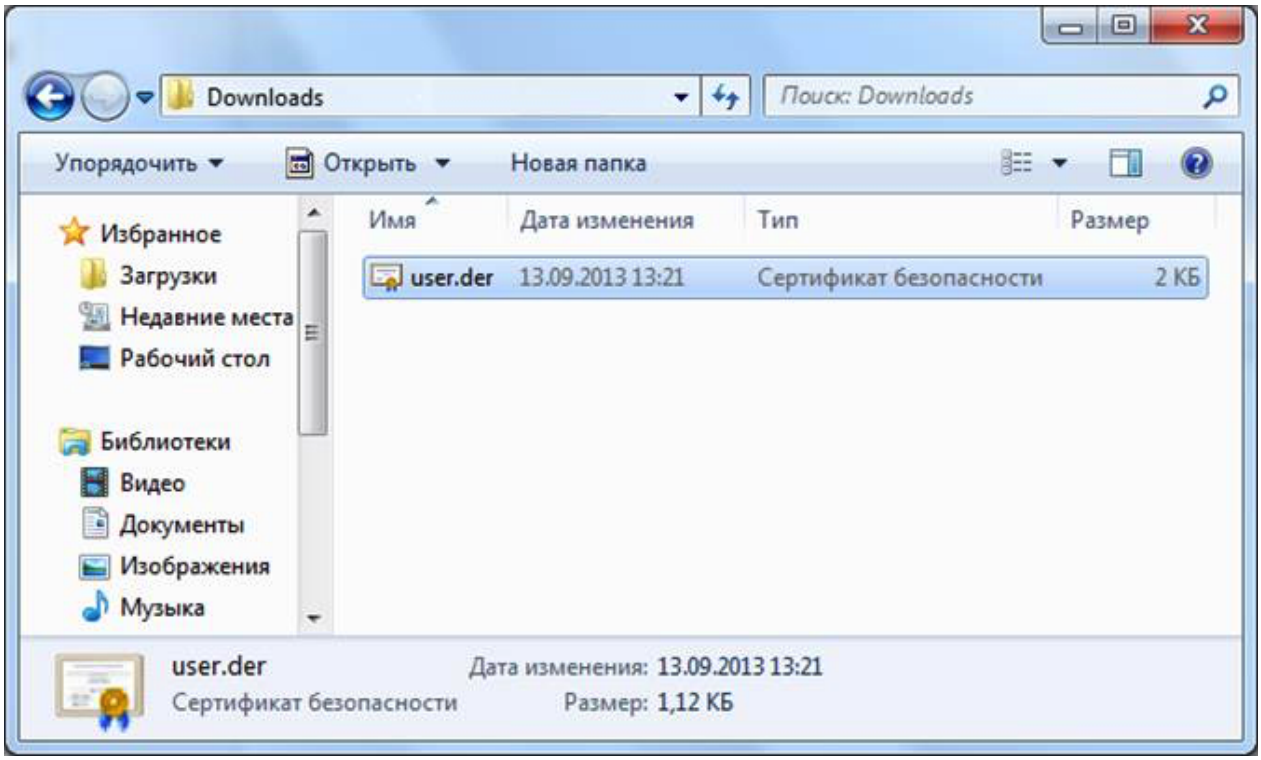


Figure 5 Selecting a certificate file

The certificate information will open. Click the **Install certificate** button:

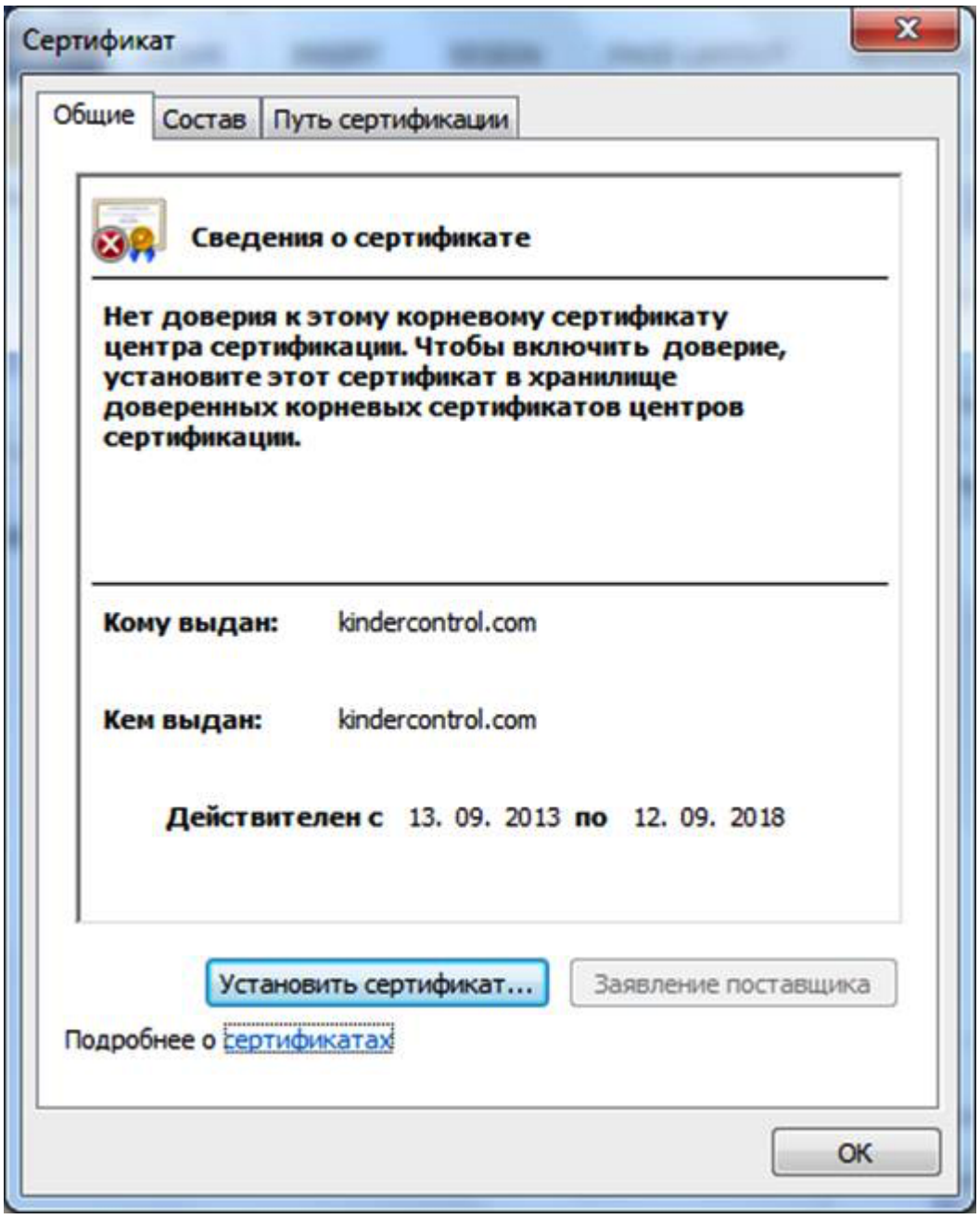


Figure 6 Certificate installation

The Certificate Import Wizard will launch. Perform the import, following all the recommendations offered by the Certificate Import Wizard:

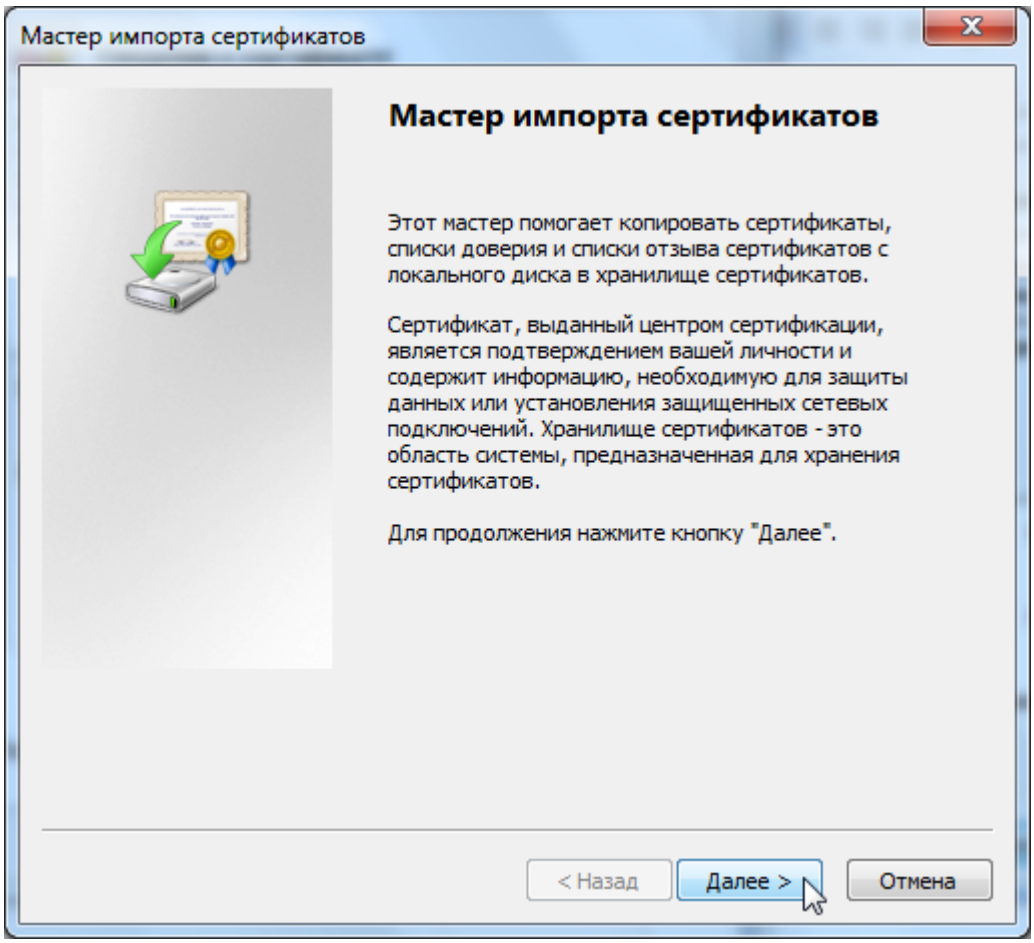


Figure 7 Certificate Import Wizard

Select the certificate store and click the **Browse** button:

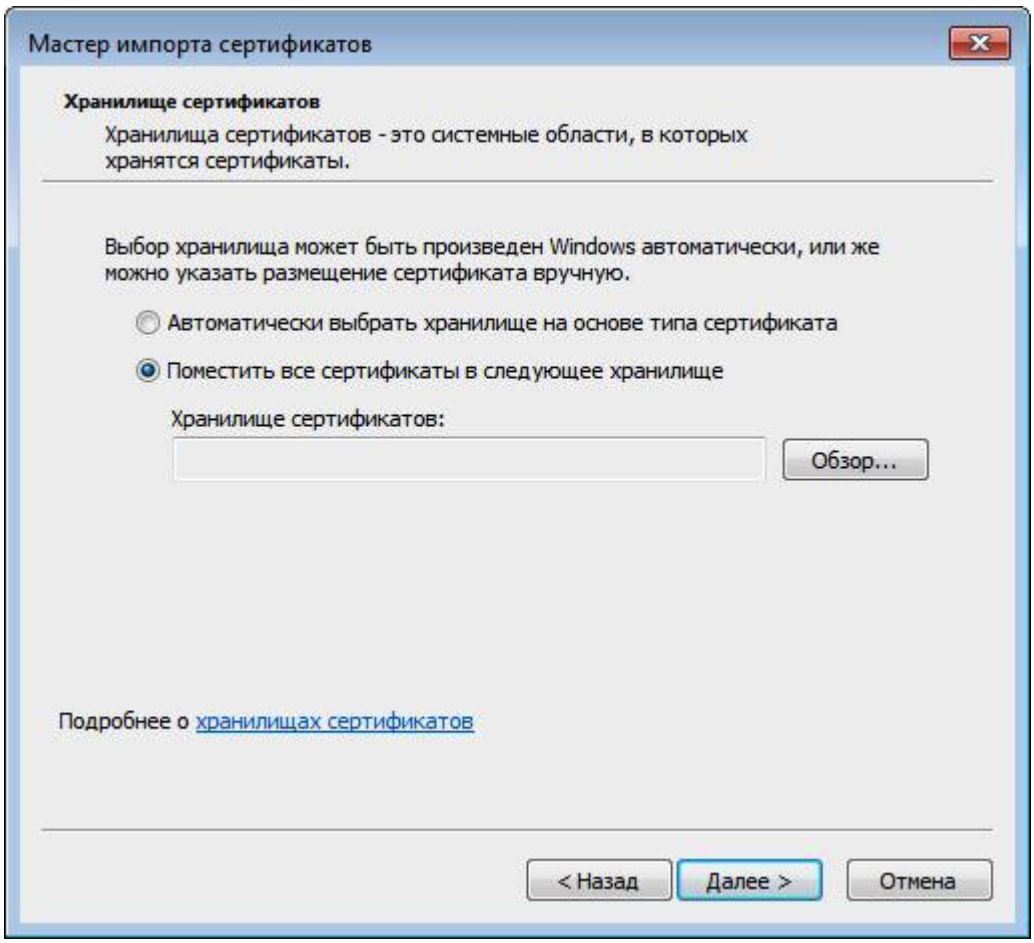


Figure 8 Selecting a storage

Select **Trusted root certification authorities** and click **OK**:

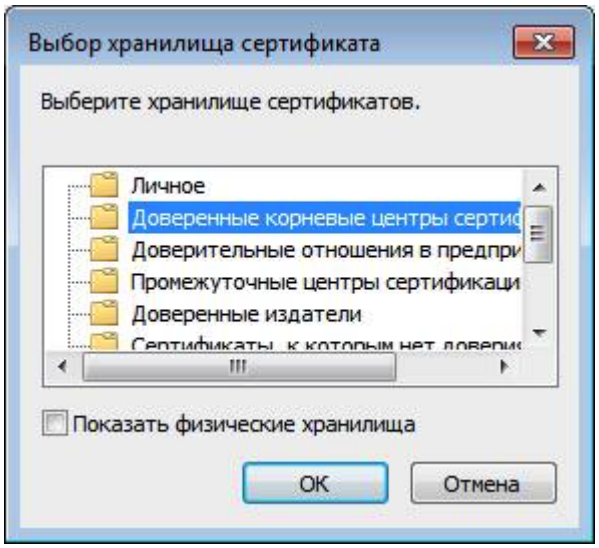


Figure 9 Selecting a storage (continued)

Click "Done":

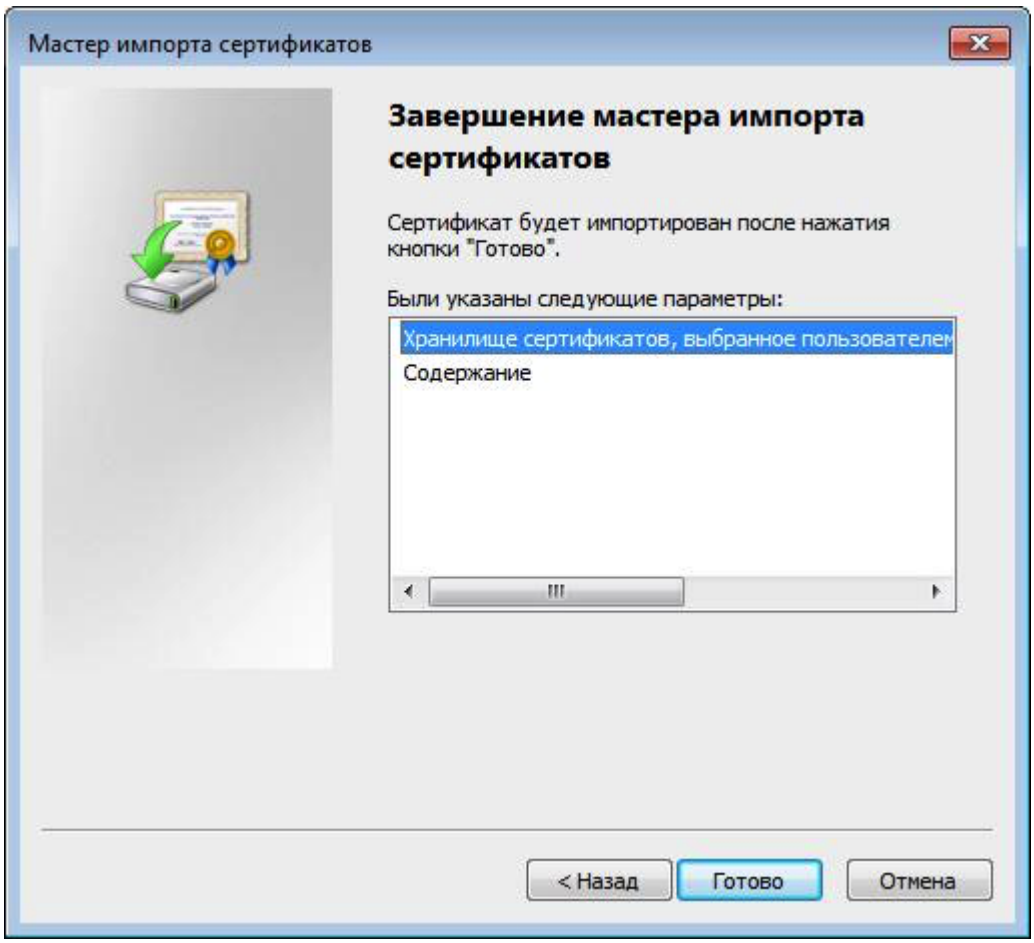


Figure 10 Import completion

When the security warning appears, click **Yes**:

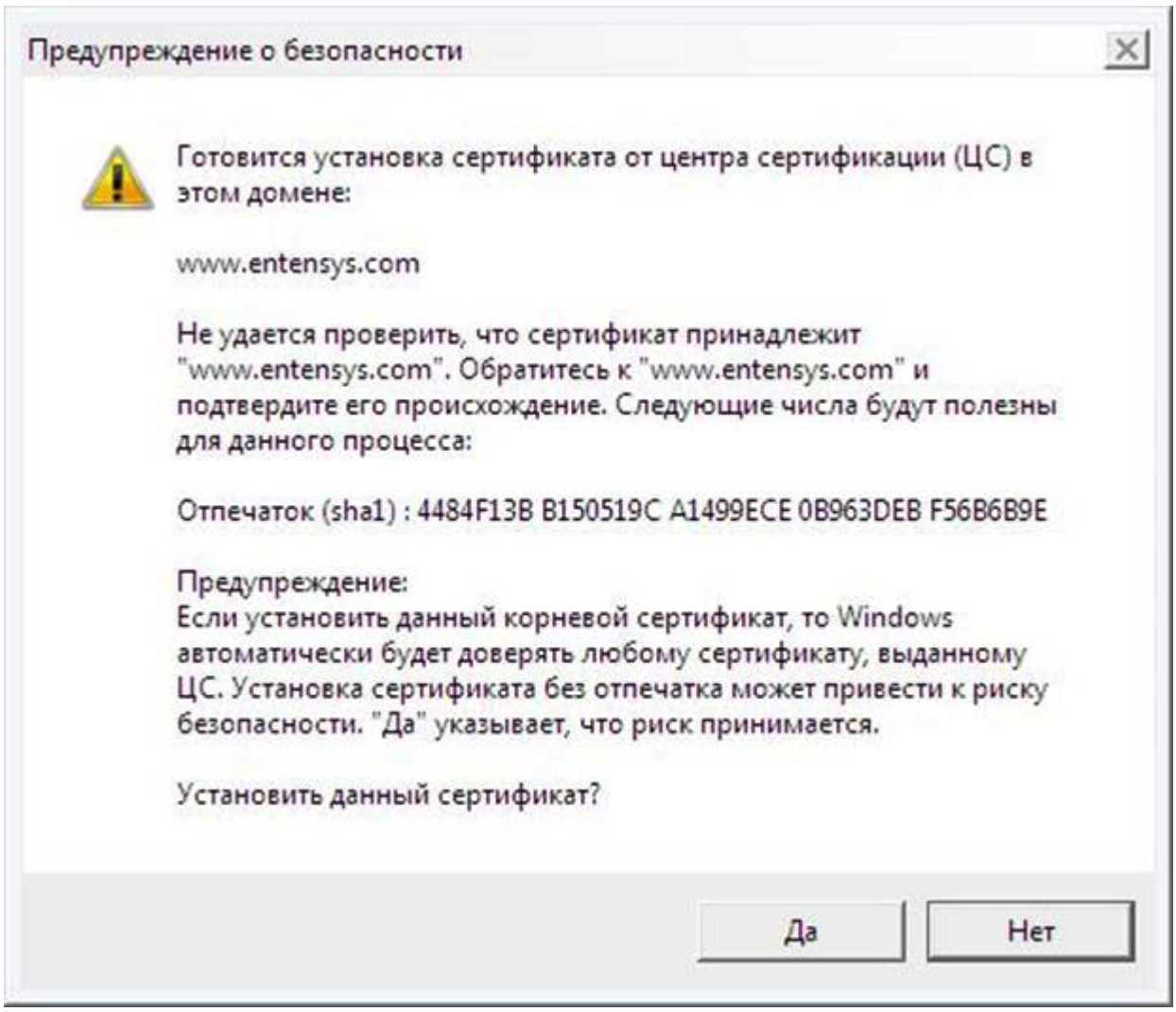


Figure 11 Consent to install a certificate

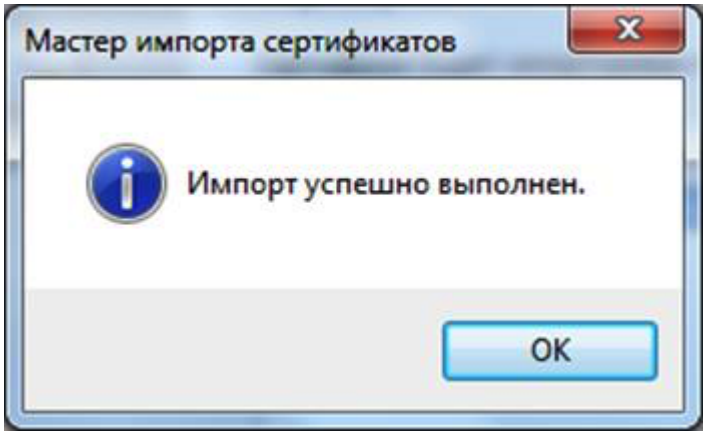


Figure 12 Installation complete

The certificate installation is complete.

Installing a Certificate in the Safari, Chrome Browser on MacOSX

Go to the folder where you downloaded the pem certificate and double click on it:

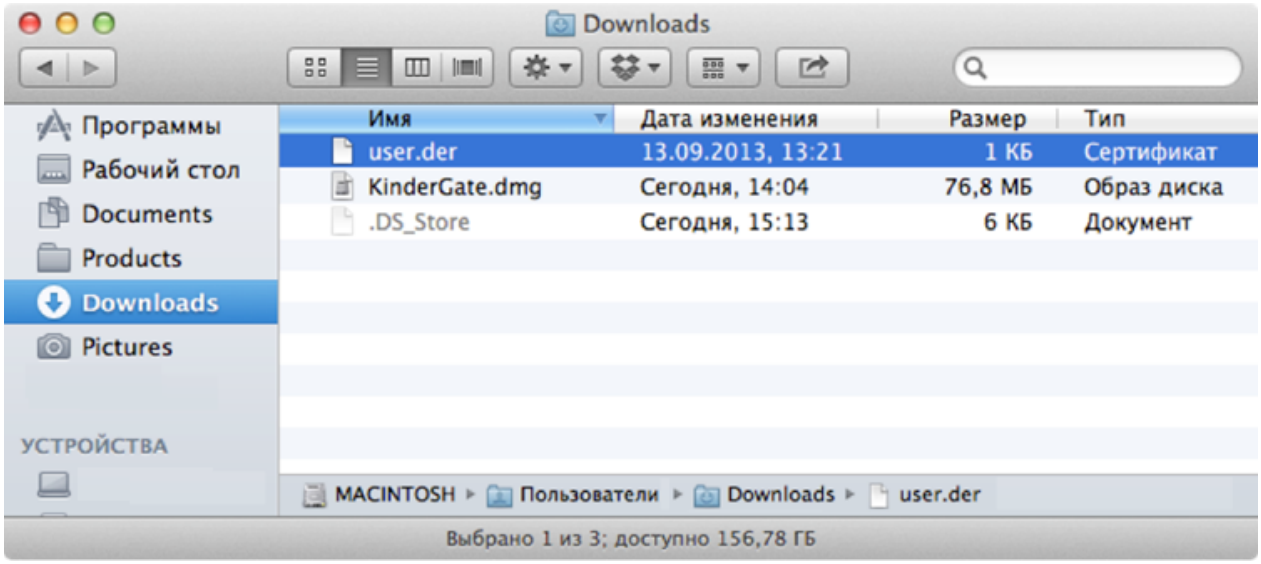


Figure 13 Selecting a certificate file

The **Keychain Access** program will start. Select **Always trust** this certificate:

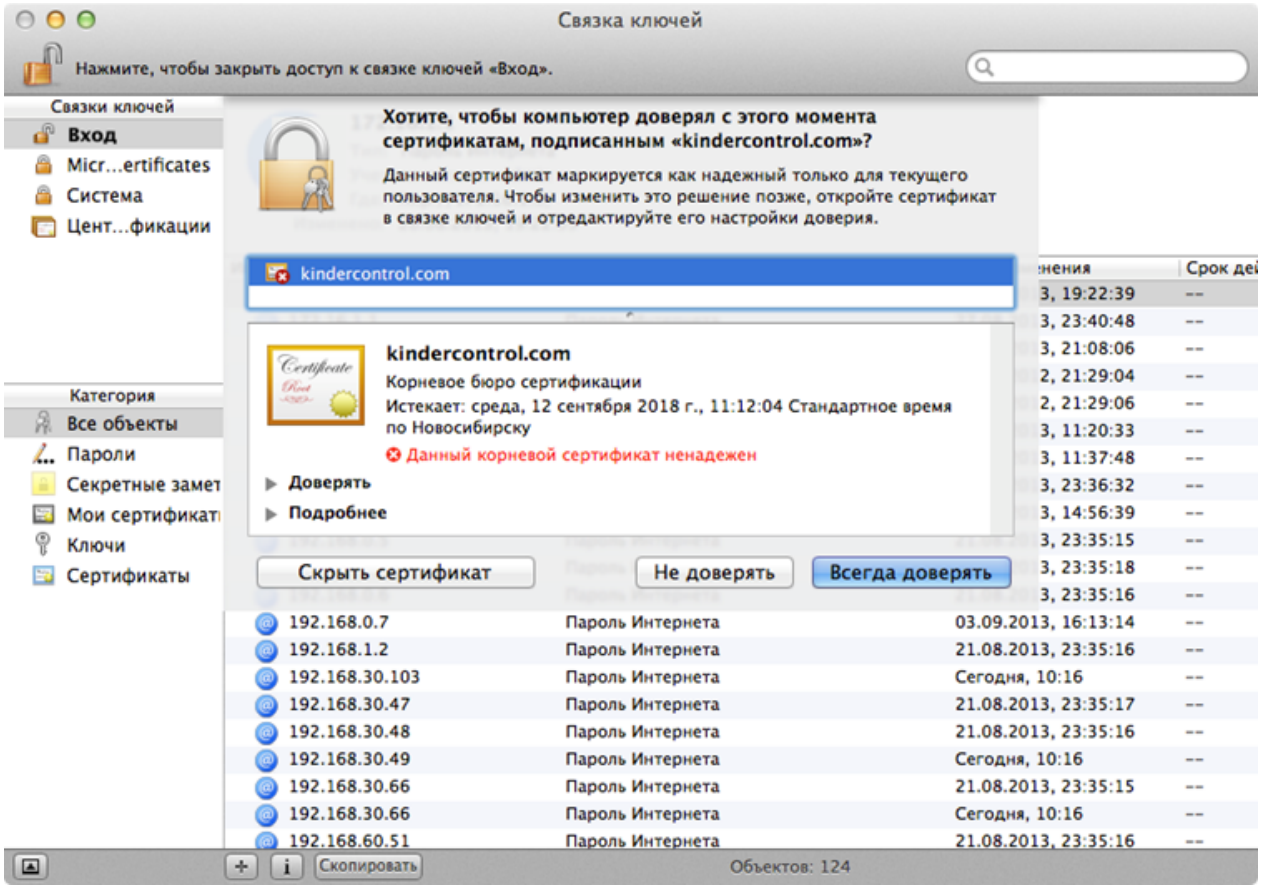


Figure 14 Certificate trust

Enter your password to confirm this operation:

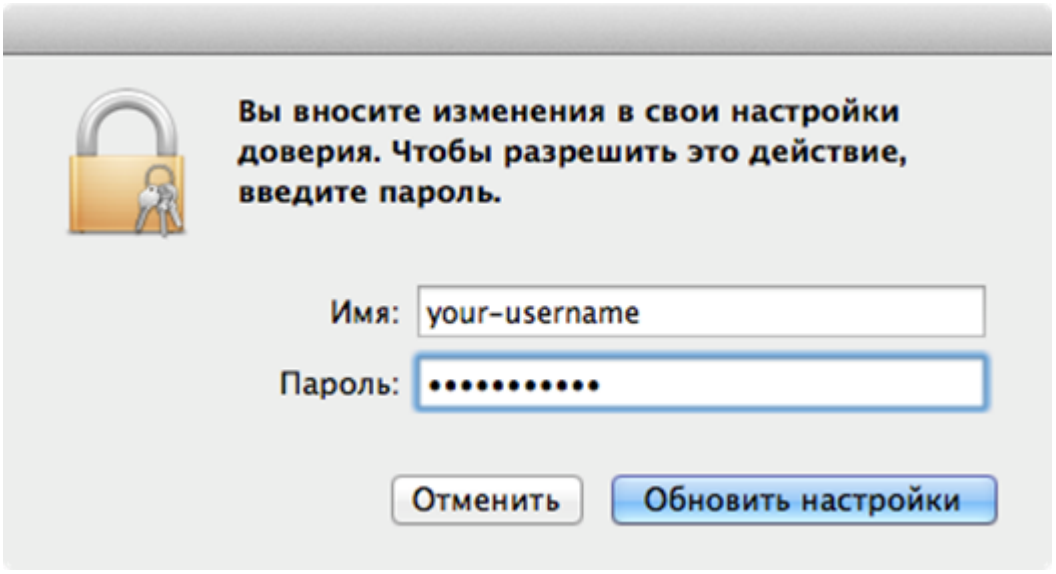


Figure 15 Password entry

The certificate is installed.

Installing a Certificate in the Firefox Browser

Installing a certificate in the Firefox browser is similar for all operating systems. Let us consider the installation using Windows OS as an example.

Open Firefox browser settings (**Tools → Options**):

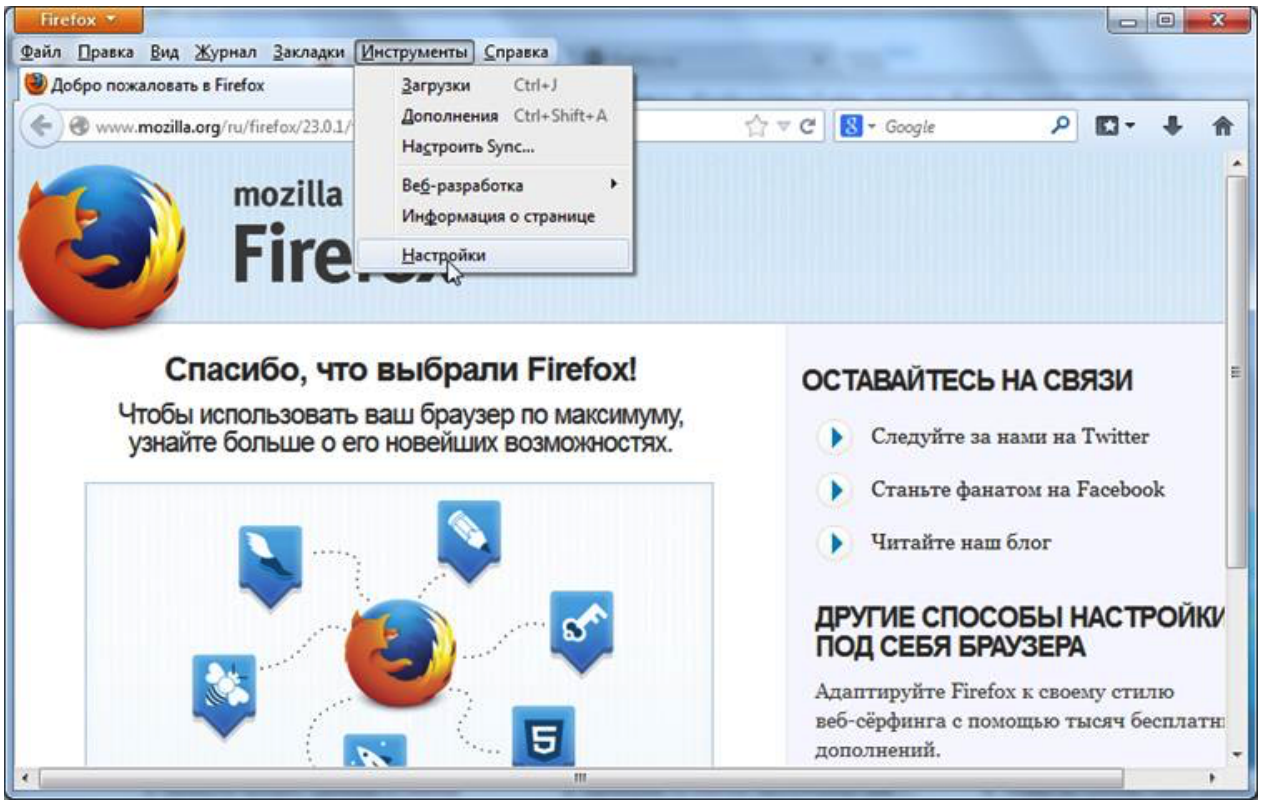


Figure 16 Entering Settings mode

Go to the **Additional** section and select the **Certificates** tab. Click the **View certificates** button:

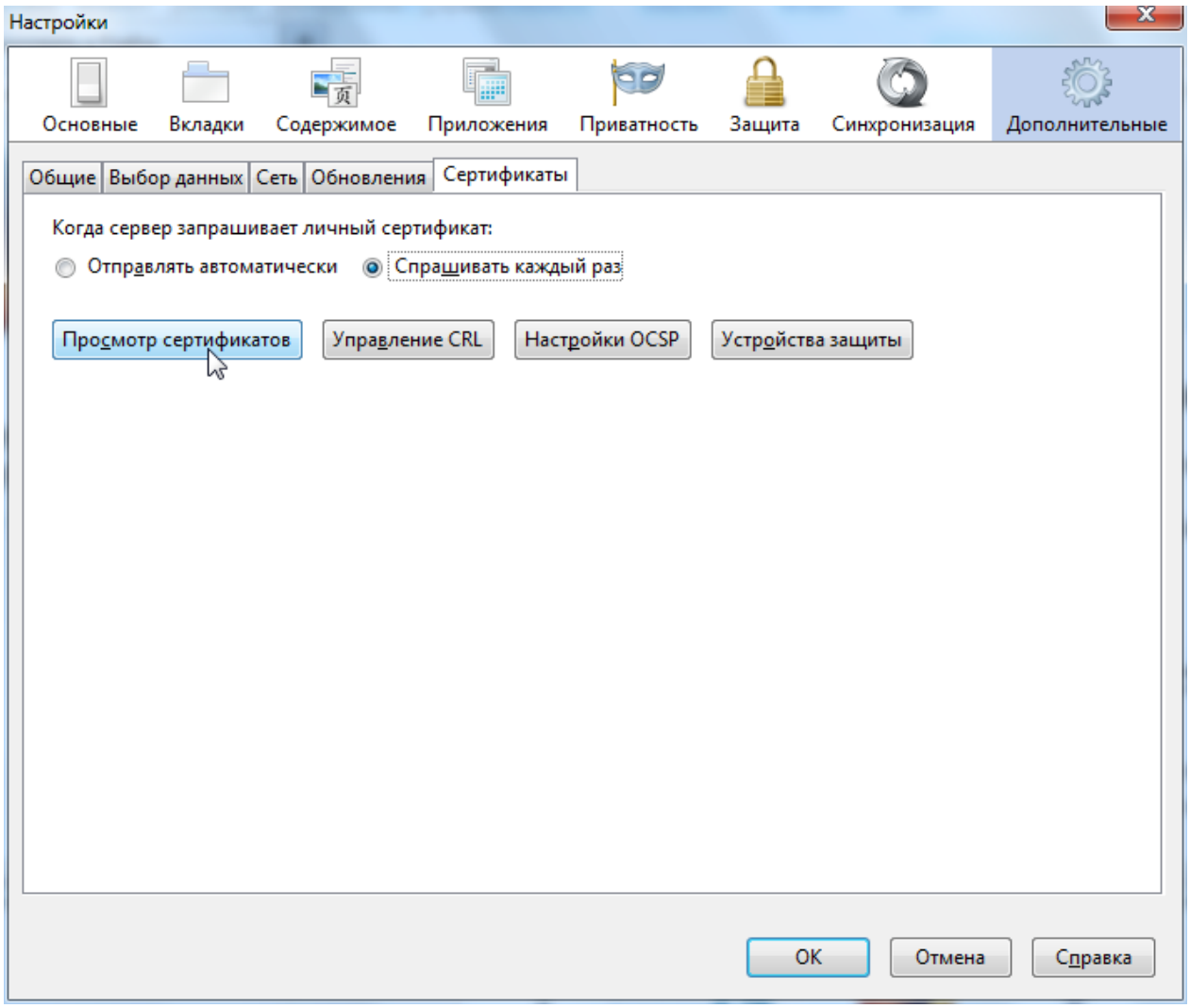


Figure 17 Certificates section

Click the **Import** button and specify the path to the downloaded pem certificate:

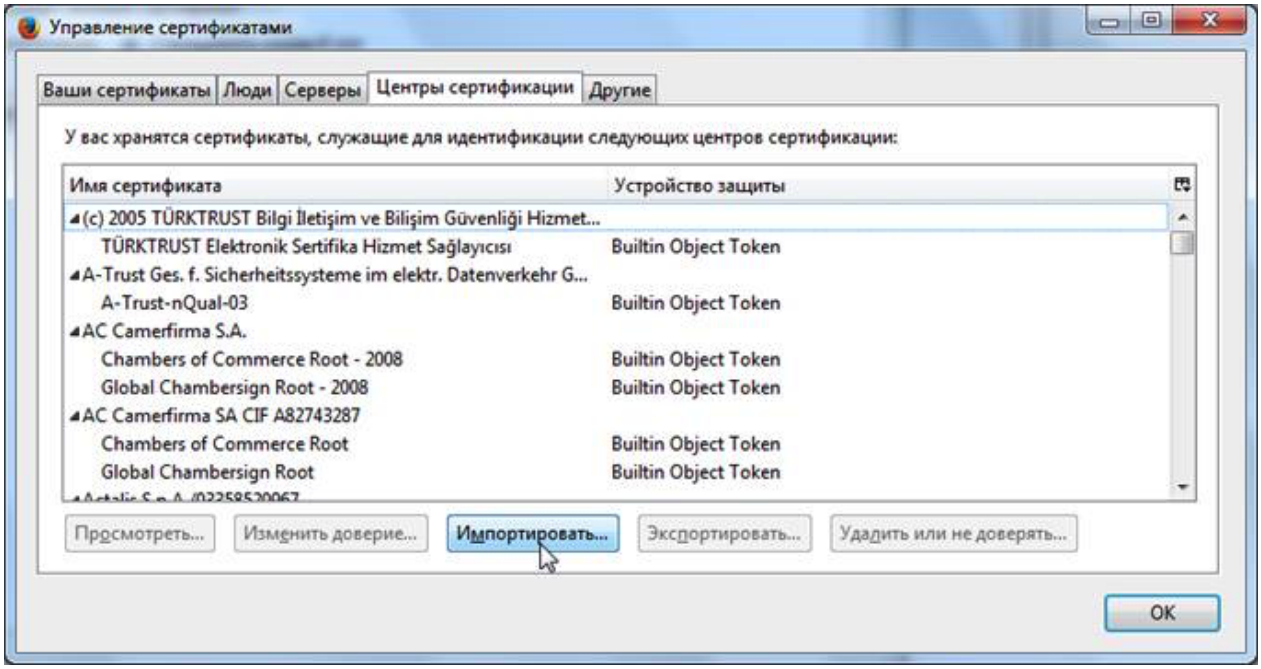


Figure 18 List of installed certificates

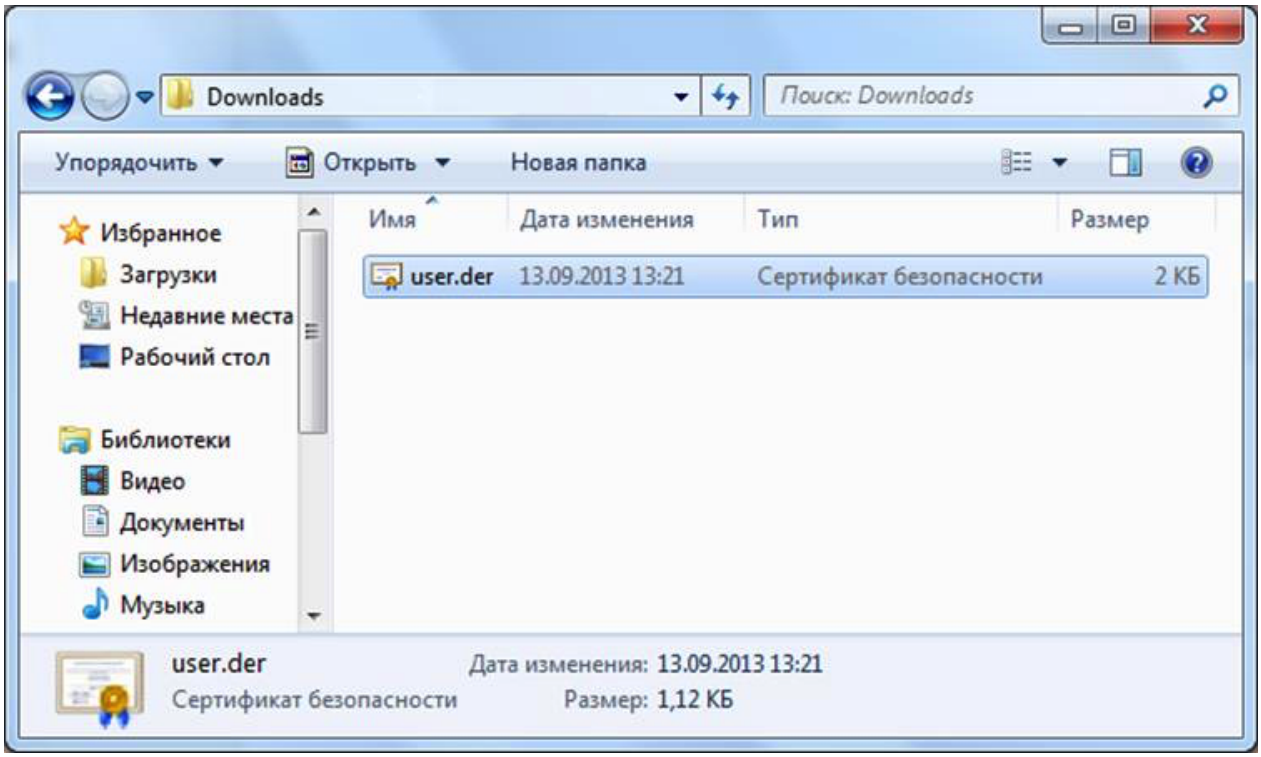


Figure 19 Selecting a certificate file

Select the **Trust when identifying websites** checkbox and click **OK**:

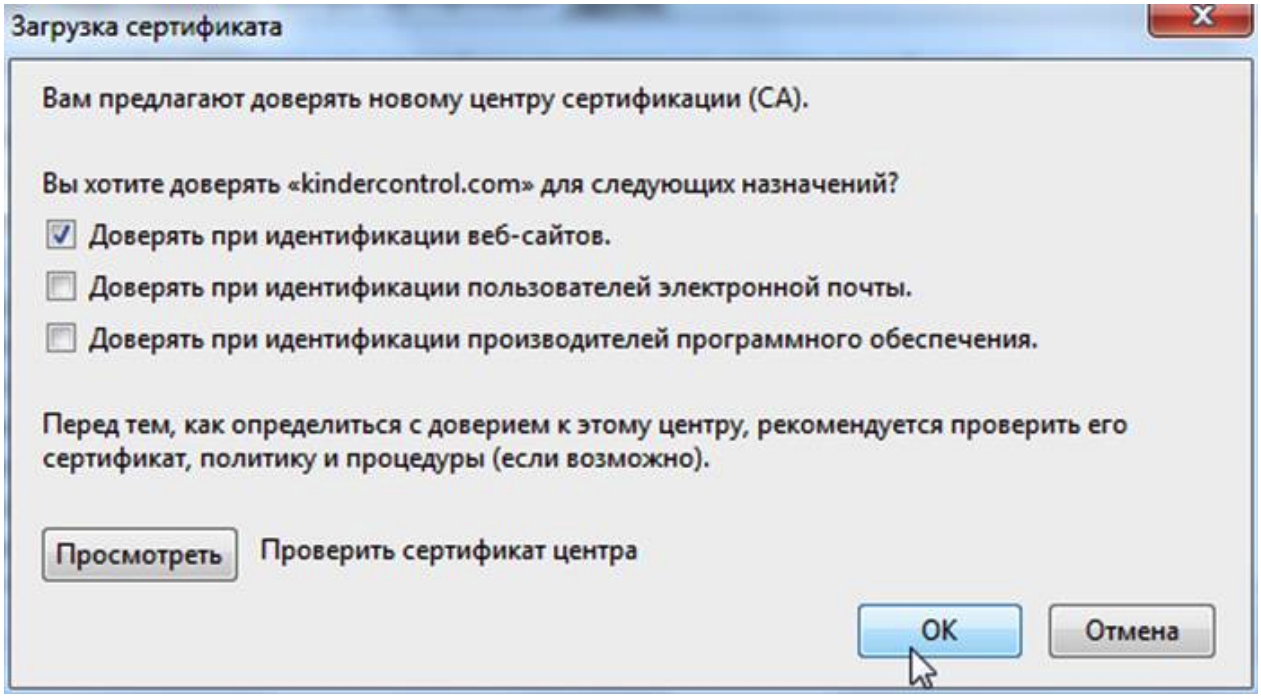


Figure 20 Selecting a trust type

The certificate installation is complete.

Examples of Certificate Generation for IKEv2 VPN

Certificate generation in Linux using OpenSSL

Example of certificate generation in Linux OS using the OpenSSL library based on a self-signed root certificate.

Actions on the VPN Server Side

1. Create the self-signed root certificate (rootCA).

```
$ openssl genrsa -aes256 -passout pass:1234 -out rootCA.key 4096
$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out
rootCA.pem -subj "/C=AR/ST=UAE/L=Dubai/O=ep.local/OU=ep.local/CN=QA"
```

here: rootCA.pem — root certificate.

Verify that the certificate is a root certificate (the output should include the following line: CA:TRUE):

```
$ openssl x509 -in rootCA.pem -text
```

2. Create a VPN server certificate based on the root certificate.

- Requirements: **key usage: server auth**
- Specify subjectAltName which is the same as the DNS name of the VPN server.

```
$ openssl genrsa -aes256 -passout pass:1234 -out server.pass.key 4096

$ openssl rsa -passin pass:1234 -in server.pass.key -out server-key.pem
```

here: server-key.pem — private key.

To generate a request for issuing the certificate, create the openssl-server.cnf file containing data for certificate request: **Example** of file with data:

```
[ req ]
prompt = no
days = 365
req_extensions = v3_req
distinguished_name = req_distinguished_name

[ req_distinguished_name ]
C = AR
ST = UAE
L = Dubai
O = ep.local
OU = ep.local
CN = vpnserver.ep.local #dns name of the vpn server
emailAddress = mail1@ep.local

[ v3_req ]
keyUsage = critical, digitalSignature
extendedKeyUsage = serverAuth
subjectAltName = @sans

[ sans ]
DNS.0 = vpnserver.ep.local # dns name of the vpn server
```

Create a request for issuing the certificate using data from the openssl-server.cnf file above. At this point the "Subject" section of the certificate is filled:

```
$ openssl req -new -key server-key.pem -out server.csr -config openssl-
server.cnf
```

Sign the request using the root certificate. At this point the "X509v3 extentions" section of the certificate is filled:

```
$ openssl x509 -CAcreateserial -req -extfile openssl-server.cnf -
extensions v3_req -days 365 -in server.csr -CA rootCA.pem -CAkey
rootCA.key -out server-cert.pem
```

where: server-cert.pem is the VPN server certificate.

3. Import the VPN server certificate in the DCFW admin console which acts as a VPN server. To do that, go to the **UserGate → Certificates** section and click **Import**. In the pop-up window specify the name of the certificate and add the generated files for the VPN server certificate (sever-cert.pem) and the private key (server-key.pem).

4. Import the root certificate in the DCFW admin console which acts as a VPN server. To do that, go to the **UserGate → Certificates** section and click **Import**. In the pop-up window specify the name of the certificate and add the generated self-signed root certificate (rootCA.pem) without the private key.

5. Create a client certificate profile in the DCFW admin console which acts as a VPN server. To do that, go to **UserGate → User certificate profiles** and click **Add**. In the opened window specify the name of the profile, add the root certificate which was imported at the previous step and select the authorization field **Common-name** or **Subject alt name** to get the username.

6. When setting VPN you will need to create a VPN security profile. Multiple security profiles may be used for connecting to different client types. To create a VPN security profile in the DCFW admin console which acts as a VPN server, go to the **VPN → Server security profiles** section and click **Add**. In the pop-up window specify the necessary security profile parameters (for more details, see VPN Settings). If the VPN is created using IKEv2 protocol, specify the VPN server certificate imported at step 3. Also add the user certificate profile created at step 5 if PKI authentication mode is used.

Actions on the VPN Client Side

1. Create the VPN client certificate.

To generate a request for issuing the VPN client certificate, create the openssl-client.cnf file containing data for certificate request: **Example** of file with data:

```
[ req ]
prompt             = no
days               = 365
req_extensions     = v3_req
distinguished_name = req_distinguished_name

[ req_distinguished_name ]
C                  = AR          #             optional parameter
```

```
ST                = UAE         #             optional parameter
L                   = Dubai       #               optional parameter
O                   = ep.local    # domain name, optional parameter
OU                  = ep.local    # domain name, optional parameter
CN                  = user1@ep.local          # required,  ID of the
user to whom the certificate was issued

[ v3_req ]
keyUsage            = critical, digitalSignature
extendedKeyUsage    = clientAuth
subjectAltName      = email:user1@ep.local # User whose account will be
used to connect to the VPN server.
```

The subjectAltName is used when the username in the client certificate profile is equal to **Subject alt name**.

Generate the private key for the VPN client:

```
$ openssl genrsa -aes256 -passout pass:1234 -out client.pass.key 4096

$ openssl rsa -passin pass:1234 -in client.pass.key -out client-key.pem
```

Create a request for issuing the certificate using data from the openssl-client.cnf file above and sign it using the root certificate:

```
$ openssl req -new -key client-key.pem -out client.csr -config openssl-
client.cnf

$ openssl x509 -CAcreateserial -req -extfile openssl-client.cnf -
extensions v3_req -days 365 -in client.csr -CA rootCA.pem -CAkey
rootCA.key -out client-cert.crt
```

3. Create the client.pfx file containing the private key and the user certificate. This file will be used by Windows clients in the Remote Access VPN scenario. This file is loaded in Windows and used to connect to the VPN.

```
$ openssl pkcs12 -export -passout pass:1234 -out client.pfx -inkey
client-key.pem -in client-cert.crt
```

4. Import the client.pfx file into Windows and place it in the **Local computer** section, **Personal** storage.

5. To use the client certificate at the node (VPN client) in the Site-to-Site VPN with IKEv2 scenario, it is necessary to import the created VPN client certificate. To do that, go to the **UserGate → Certificates** section in the DCFW admin console which acts as a VPN client and click **Import. In the pop-up window specify the name of the certificate and add the generated files for the VPN client certificate (client-cert.crt) and the private key (client-key.pem). Next when creating a VPN security profile at the stage of VPN configuration go to the VPN → Client security profiles section and click Add. In the pop-up window specify the necessary security profile parameters (for more details, see VPN Settings). If the VPN is created using IKEv2 protocol, specify the previously imported VPN client certificate in the "Client certificate" field.**

Certificate Generation by the Microsoft Server Certification Center

Examples of certificate generation by the Microsoft Server Certification Center for the Remote Access VPN scenario.

1. Issue the root certificate (rootCA) using the Certification Center.

2. Issue the VPN server certificate based on the root certificate (rootCA).

- Specify the requirements: **key usage: server auth**
- Minimum key size: **4096**
- Specify **subjectAltName which is the same as the DNS name of the VPN server.**
- Create the template for issuing user certificates. The UPN user attribute should match the CN and/or SAN:principal name certificate attributes.

Actions on the VPN Server Side

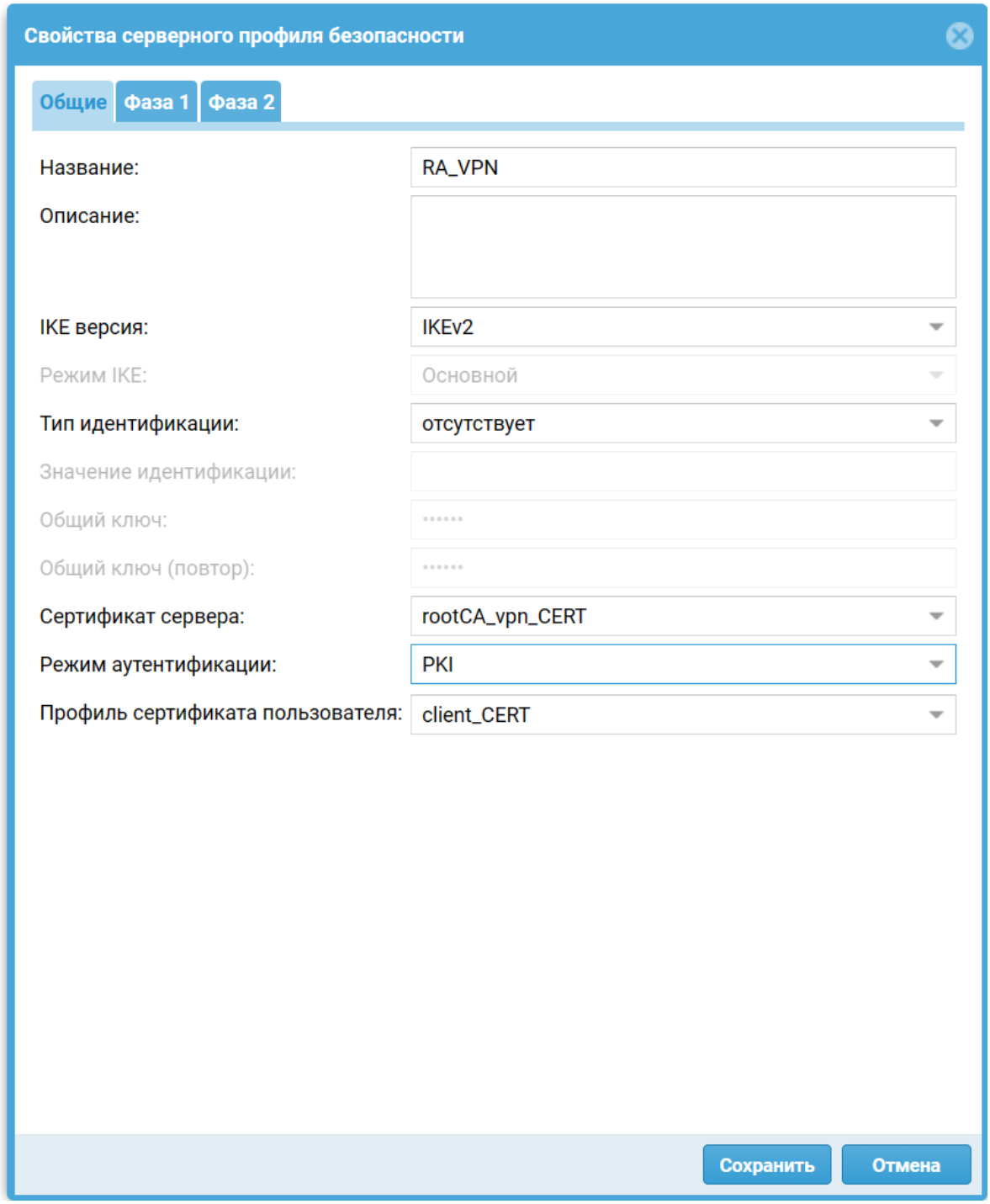
1. Import the root certificate (rootCA) in the DCFW admin console which acts as a VPN server. To do that, go to the **UserGate → Certificates** section and click **Import**.

2. Import the VPN server certificate in the DCFW admin console which acts as a VPN server. To do that, go to the UserGate → Certificates section and click Import.

3. Create a client certificate profile in the DCFW admin console which acts as a VPN server. To do that, go to **UserGate → User certificate profiles** and click **Add**. In the opened window specify the name of the profile, add the root certificate which was imported earlier and select the authorization field **Common-name** or **Subject alt name** to get the username.

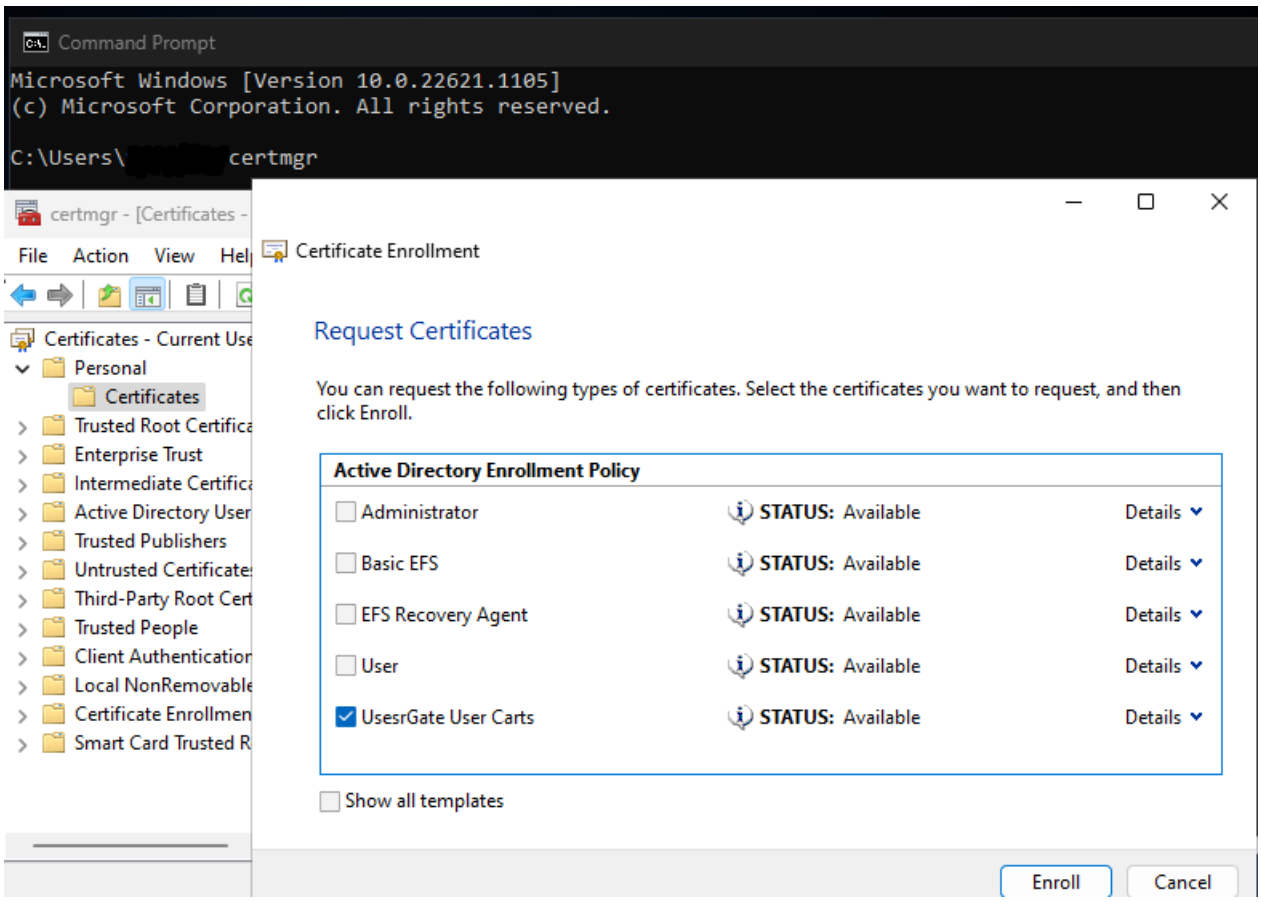
4. In the **VPN → Server security profiles** section open the **Remote access VPN profile** and add:

- The certificate to the **Server certificate** field.
- Select the created **User certificate profile**.
- For the **Authentication mode** parameter set using **PKI** certificates.

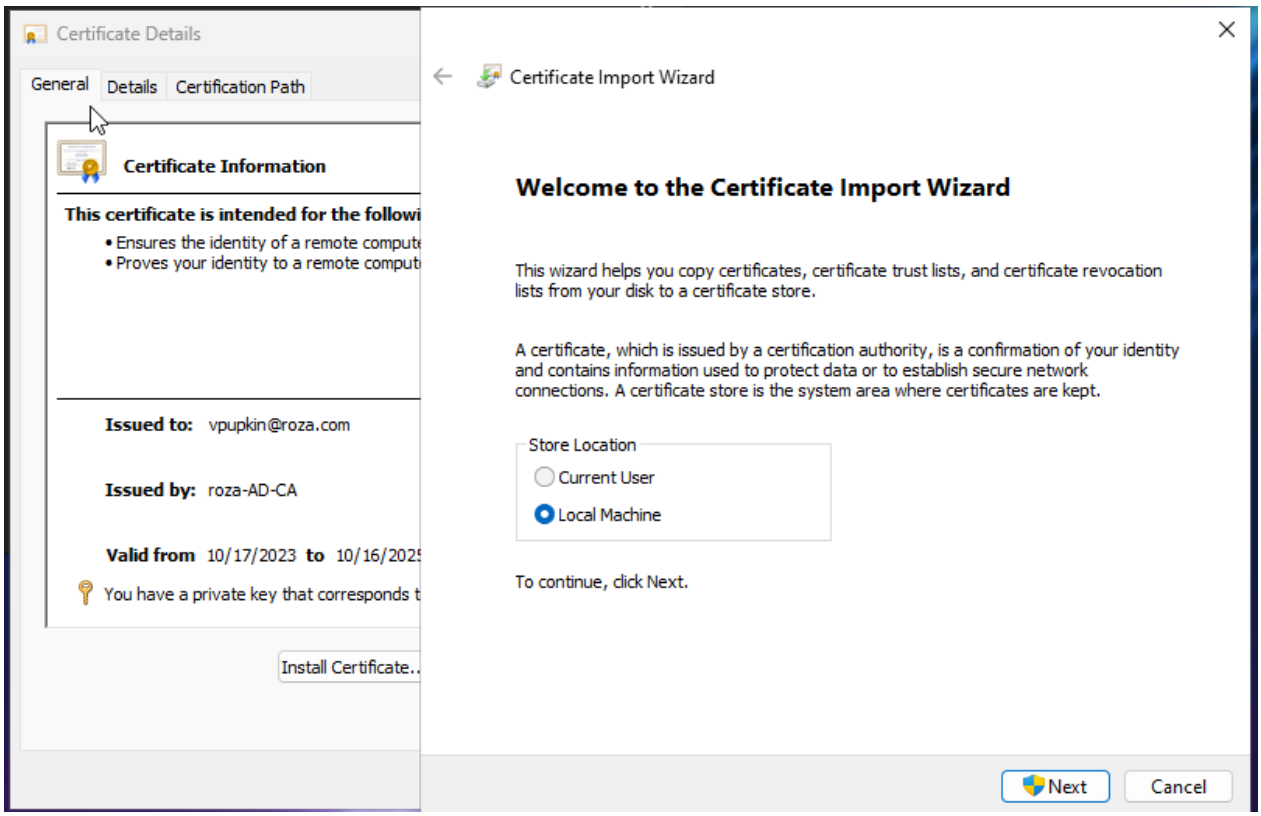


Actions on the VPN Client Side

1. Request the certificate at the client computer in accordance with the previously created user certificate template.



2. Get the certificate and put it in the Personal storage at the Local computer repository.


Certificate Details
General
Details
Certification Path
Certificate Information
This certificate is intended for the followi
• Ensures the identity of a remote compute
• Proves your identity to a remote comput
Issued to: vpupkin@roza.com
Issued by: roza-AD-CA
Valid from 10/17/2023 to 10/16/2025
You have a private key that corresponds t
Install Certificate...
Certificate Import Wizard
Welcome to the Certificate Import Wizard
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.
A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.
Store Location
Current User
Local Machine
To continue, click Next.
Next
Cancel