

Table of Contents

Table of Contents	1
UserGate Policy Language (UPL)	7
Общие сведения	8
Условия	11
Встроенные библиотеки	17
Определения	18
Свойства	19
<code>block_invalid_cert</code> Блокирование сайтов с некорректными сертификатами. Действительно только для правил инспектирования SSL. Синтаксис: <code>InvalidCertificate ::= block_invalid_cert (' boolean ') boolean ::= yes no true false</code> <code>check_revoc_cert</code> Проверка по списку отозванных сертификатов. Действительно только для правил инспектирования SSL. Синтаксис: <code>ChekRevocation ::= check_revoc_cert (' boolean ') boolean ::= yes no true false</code> <code>block_expired_cert</code> Блокировка сертификатов с истекшим сроком действия. Действительно только для правил инспектирования SSL. Синтаксис: <code>ExpiredCertificate ::= block_expired_cert (' boolean ') boolean ::= yes no true false</code> <code>block_self_signed_cert</code> Блокировка самоподписанных сертификатов. Действительно только для правил инспектирования SSL. Синтаксис: <code>SelfSignedCertificate ::= block_self_signed_cert (' boolean ') boolean ::= yes no true false</code> <code>ssl_profile</code> Профиль SSL. Действительно только для правил инспектирования SSL, reverse-прокси, веб-портала. Синтаксис: <code>SslProfile ::= ssl_profile (' string word ')</code> <code>ssl_forward_profile</code> Профиль пересылки SSL. Действительно только для правил инспектирования SSL. Синтаксис: <code>SslForwardProfile ::= ssl_forward_profile (' string word ')</code> <code>profile_id</code> Свойства правил пропускной способности <code>bandwidth_pool</code> Полоса пропускания. Действительно только для правил пропускной способности. Синтаксис: <code>BandwidthPool ::= bandwidth_pool (' bandwidth ') bandwidth ::= "100 Kbps" "512 kbps" "1 Mbps" "2 Mbps" "5 Mbps" "10 Mbps" "20 Mbps" "50 Mbps" "100 Mbps"</code> Пример: <code>scenario = "Сценарий обнаружения torrent" bandwidth_pool("1 Mbps") enable(true) name("Пропускная способность torrent")</code> <code>mark</code> Свойства правил защиты почтового трафика Маркировка. Текст тега, который маркирует письмо. Действительно только для правил защиты почтового трафика. Синтаксис: <code>Mark ::= mark (' string word ')</code> <code>mark_hdr</code> Заголовок. Поле, куда помещается тег маркировки. Действительно только для правил защиты почтового трафика. Синтаксис: <code>MarkHeader ::= mark_hdr (' word ')</code> <code>antispam_kav</code> Проверка антиспамом. Действительно только для правил защиты почтового трафика. Синтаксис: <code>AntispamKav ::= antispam_kav (' boolean ') boolean ::= yes no true false</code> <code>antispam_usergate</code> Проверка антиспамом UserGate. Действительно только для правил защиты почтового трафика. Синтаксис: <code>AntispamUsergate ::= antispam_usergate (' boolean ') boolean ::= yes no true false</code> Пример: <code>DENY("with error") envelop_to = UserGate antispam_usergate(yes) dnsbl</code> <code>DNSBL</code> DNSBL-проверка (только SMTP). Действительно только для правил защиты почтового трафика. Синтаксис: <code>DNSBLacklistCheck ::= dnsbl (' boolean ') boolean ::= yes no true false</code> Пример: <code>DENY service = SMTP dnsbl(yes)</code> <code>target_ip</code> Свойства правил NAT и маршрутизации Адрес назначения DNAT, если задано действие <code>dnat</code> или <code>port_mapping</code>. Новая IP-сеть/маска, если задано действие <code>netmap</code>. Действительно только для правил NAT и маршрутизации. Синтаксис: <code>TargetIp ::= target_ip (' ipv4 ipv4_with_mask ')</code> Пример: <code>PASS target_ip("192.168.1.20") dnat</code> <code>target_snat</code> Включение SNAT. При включении данной опции UserGate будет изменять адрес источника в пакетах из внешней сети на свой IP-адрес. Действительно только для правил NAT и маршрутизации. Синтаксис: <code>TargetSnat ::= target_snat (' boolean ') boolean ::= yes no true false</code> <code>snat_target_ip</code> SNAT IP (внешний адрес). Явно указывает IP-адрес, на который будет заменен адрес источника при подмене адресов пакетов. Действительно только для правил NAT и маршрутизации. Синтаксис: <code>SnatTargetIp ::= snat_target_ip (' ip_address ')</code> <code>port_map</code> Порт-форвардинг. Переопределение портов публикуемых сервисов. Действительно только для правил NAT	

и маршрутизации. Синтаксис: PortMap ::= port_map '(' protocol, port_from, port_to ')' protocol ::= tcp | udp | smtp | smpts port_from ::= integer port_to ::= integer port_from — номер TCP/UDP-порта, на который пользователи шлют запросы; port_to — номер TCP/UDP-порта, на который будут пересылаться запросы пользователей на внутренний публикуемый сервер. direction Направление подмены сетей. Действительно только для правил NAT и маршрутизации. Синтаксис: Direction ::= direction '(' input | output ')' input — входящий, подменяется IP-сеть назначения. Будут изменены IP-адреса назначения в трафике; output — исходящий, подменяется IP-сеть источника. Будут изменены IP-адреса источника в трафике. Свойства правил DNS-прокси dns_server Список IP-адресов DNS-серверов. Действительно только для правил DNS-прокси. Синтаксис: DnsServer ::= dns_server '(' ip_address | ip_address_list ')' Свойства правил reverse-прокси cert_auth_enabled Аутентификация по сертификату. Действительно только для правил Reverse-прокси. Синтаксис: CertAuthEnabled ::= cert_auth_enabled '(' boolean ')' boolean ::= yes | no | true | false is_https Включение поддержки HTTPS. Действительно только для правил Reverse-прокси. Синтаксис: IsHttps ::= is_https '(' boolean ')' boolean ::= yes | no | true | false rewrite_path Подмена путей. Действительно только для правил Reverse-прокси. Синтаксис: RewritePath ::= rewrite_path '(' path_from, path_to ')' path_from — изменить с (домен и/или путь URL, которые требуется изменить); path_to — изменить на (домен и/или путь URL, на которые требуется заменить старые). waf_profile Профиль WAF. Действительно только для правил Reverse-прокси. Синтаксис: WafProfile ::= waf_profile '(' string | word | list ')' Свойства правил веб-портала icon Иконка, которая будет отображаться на веб-портале для данной закладки. Действительно только для правил веб-портала. Синтаксис: Icon ::= icon '(' string | word ')' additional_url Вспомогательные URL, необходимые для работы основного URL, но которые нет необходимости публиковать для пользователей. Действительно только для правил веб-портала. Синтаксис: AdditionalUrl ::= additional_url '(' string | word | list ')' rdp_check_session_alive Проверка авторизации для RDP-сессий. Действительно только для правил веб-портала. Синтаксис: RdpCheckSessionAlive ::= rdp_check_session_alive '(' boolean ')' boolean ::= yes | no | true | false transparent_auth Включение прозрачной аутентификации пользователя. Действительно только для правил веб-портала. Синтаксис: TransparentAuth ::= transparent_auth '(' boolean ')' boolean ::= yes | no | true | false Свойства правил веб-безопасности (Safe browsing) enable_adblock Блокировка рекламы. Синтаксис: EnableAdblock ::= enable_adblock '(' boolean ')' boolean ::= yes | no | true | false url_list_exclusions URL-список сайтов исключений, для которых блокировать рекламу не требуется. Синтаксис: UrlListExclusions ::= url_list_exclusions '(' list_libs ')' list_libs ::= lib_name ',' list_libs lib_name ::= word | string enable_injector Позволяет вставить произвольный код во все веб-страницы. Синтаксис: EnableInjector ::= enable_injector '(' boolean ')' boolean ::= yes | no | true | false custom_injector Код инжектора. Синтаксис: CustomInjector ::= custom_injector '(' string ')' safe_search Функция безопасного поиска. Синтаксис: SafeSearch ::= safe_search '(' boolean ')' boolean ::= yes | no | true | false search_history_logging Запись в журнал поисковых запросов пользователей. Синтаксис: SearchHistoryLogging ::= search_history_logging '(' boolean ')' boolean ::= yes | no | true | false social_sites_block Блокировка приложения в популярных социальных сетях. Синтаксис: SocialSitesBlock ::= social_sites_block '(' boolean ')' boolean ::= yes | no | true | false Свойства BYOD max_device_number Максимальное количество устройств, с которых пользователь может получать доступ в сеть. Действительно только для правил BYOD. Синтаксис: MaxDeviceNumber ::= max_device_number '(' integer ')' max_active_device_number Максимальное количество устройств, с которых пользователь одновременно может получать доступ в сеть. Действительно только для правил BYOD. Синтаксис: MaxActiveDeviceNumber ::= max_active_device_number '(' integer ')' device_type Тип устройств, для которых применяется данное правило политики BYOD. Действительно только для правил BYOD. Синтаксис: DeviceType ::= device_type '(' word | string | list ')' approving_required Подтверждение администратора. Действительно только

для правил BYOD. Синтаксис: ApprovingRequired ::= approving_required '(' boolean ')' boolean ::= yes | no | true | false Свойства инспектирования SSH block_ssh_shell Блокировка удалённого запуска shell. Действительно только для правил инспектирования SSH. Синтаксис: BlockSshShell ::= block_ssh_shell '(' boolean ')' boolean ::= yes | no | true | false block_ssh_exec Блокировка удалённого выполнения команд по SSH. Действительно только для правил инспектирования SSH. Синтаксис: BlockSshExec ::= block_ssh_exec '(' boolean ')' boolean ::= yes | no | true | false block_sftp Блокировка соединения SFTP (Secure File Transfer Protocol). Действительно только для правил инспектирования SSH. Синтаксис: BlockSftp ::= block_sftp '(' boolean ')' boolean ::= yes | no | true | false ssh_command Команда linux, которую требуется передать, в формате ssh user@host 'command'. Действительно только для правил инспектирования SSH. Синтаксис: SshCommand ::= ssh_command '(' string ')' Свойства правил VPN-сервера auth_profile Профиль аутентификации. Действительно только для правил VPN-сервера. Синтаксис: AuthProfile ::= auth_profile '(' string | word ')' vpn_network Сеть VPN. Действительно только для правил VPN-сервера. Синтаксис: VpnNetwork ::= vpn_network '(' string ')' interface Интерфейс VPN. Действительно только для правил VPN-сервера и VPN-клиента. Синтаксис: Interface ::= interface '(' word ')' Свойства правил VPN-клиента server_address IP-адрес сервера. Действительно только для правил VPN-клиента. Синтаксис: ServerAddress ::= server_address '(' ip_address ')' password Пароль. Действительно только для правил VPN-клиента. Синтаксис: Password ::= password '(' word ')' last_error Последняя ошибка VPN. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента. Синтаксис: LastError ::= last_error '(' string ')' connection_time Время соединения. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента. Синтаксис: ConnectionTime ::= connection_time '(' word ')' status Статус. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента. Синтаксис: Status ::= status '(' string ')' Свойства правил сценариев trigger Тип срабатывания сценария. Действительно только для правил сценариев. Синтаксис: Trigger ::= trigger '(' trigger_type ')' trigger_type ::= all_users | one_user duration Продолжительность. Время в минутах, в течение которого сценарий будет активным после его активации. Действительно только для правил сценариев. Синтаксис: Duration ::= duration '(' integer ')' operation_mode Сценарий срабатывает при совпадении одного или всех условий. Синтаксис: OperationMode ::= operation_mode '(' mode ')' mode ::= all | any Свойства условий сценария scond_type Тип условий сценария. Действительно только для условий сценария. Синтаксис: SCondType ::= scond_type '(' type ')' SCondType ::= scond_type '(' type ')' count_interval Количество срабатываний. Действительно только для условий сценария "Категория URL" (url_category), "Приложение" (app), "Тип контента" (mime_type) и "Проверка состояния" (health_check). Синтаксис: CountInterval ::= count_interval '(' integer ')' max_event_count Интервал времени в минутах, за который было обнаружено count_interval срабатываний. Действительно только для условий сценария "Категория URL" (url_category), "Приложение" (app), "Тип контента" (mime_type) и "Проверка состояния" (health_check). Синтаксис: MaxEventCount ::= max_event_count '(' integer ')' packet_size Размер пакета. Размер пакета в трафике пользователя превысил указанное значение. Действительно только для условий сценария "Размер пакета" (net_packet_size). Синтаксис: PacketSize ::= packet_size '(' size ')' size ::= integer | integer KB | integer MB | integer GB 64 — размер в байтах; 2MB — размер в мегабайтах. traffic_limit Объем трафика. Действительно только для условий сценария "Ограничение трафика" (traffic). Синтаксис: TrafficLimit ::= traffic_limit '(' size ')' size ::= integer | integer KB | integer MB | integer GB period Период времени. Действительно только для условий сценария "Ограничение трафика" (traffic). Синтаксис: Period ::= period '(' time_period ')' time_period ::= minute | hour | day | week | month ips_tl Уровень угрозы COB. Действительно только для условий сценария "COB" (ips). Синтаксис: 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 Метод

проверки. Действительно только для условий сценария "Проверка состояния" (health_check). Синтаксис: CheckMethod ::= health_check_method '(' method ')' method ::= ping | dns | get health_result Результат. Действительно только для условий сценария "Проверка состояния" (health_check). Синтаксис: esult ::= health_result '(' result ')' result ::= positive | negative health_request_timeout Тайм-аут подключения (сек). Действительно только для условий сценария "Проверка состояния" (health_check). Синтаксис: RequestTimeout ::= health_request_timeout '(' integer ')' health_answer_timeout Тайм-аут ответа (сек). Действительно только для условий сценария "Проверка состояния" (health_check), "Метод проверки HTTP GET" (get). Синтаксис: ut ::= health_answer_timeout '(' integer ')' health_type_request Тип DNS-запроса. Действительно только для условий сценария "Проверка состояния" (health_check), "метод проверки DNS" (dns). Синтаксис: TypeRequest ::= health_type_request '(' type_name ')' type_name ::= word | string Свойства правил балансировки нагрузки scheduler Метод балансировки. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: 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 Реальные серверы. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: RealServer ::= real_server '(' mode, ipv4[:port], weight ')' mode ::= gate | masq | masq_snat port ::= integer weight ::= integer gate — шлюз; masq — маскрадинг; masq_snat — маскрадинг с подменой IP-адреса источника (SNAT). Пример: 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 Аварийный режим сервера. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: IpvsFallback ::= ipvs_fallback '(' mode, ipv4[:port] ')' mode ::= gate | masq | masq_snat port ::= integer monitor_kind Режим. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorKind ::= monitor_kind '(' kind ')' kind ::= ping | connect | negotiate monitor_service Сервис. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorService ::= monitor_service '(' service ')' service ::= http | dns monitor_request Запрос. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorRequest ::= monitor_request '(' string ')' monitor_response Ожидаемый ответ. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorResponse ::= monitor_response '(' string ')' monitor_interval Интервал проверки. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorInterval ::= monitor_interval '(' integer ')' monitor_timeout Время ожидания. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorTimeout ::= monitor_timeout '(' integer ')' monitor_failurecount Число неудачных попыток. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorFailurecount ::= monitor_failurecount '(' integer ')'

26

max_device_number Максимальное количество устройств, с которых пользователь может получать доступ в сеть. Действительно только для правил BYOD. Синтаксис: MaxDeviceNumber ::= max_device_number '(' integer ')' max_active_device_number Максимальное количество устройств, с которых пользователь одновременно может получать доступ в сеть. Действительно только для правил BYOD. Синтаксис: MaxActiveDeviceNumber ::= max_active_device_number '(' integer ')' device_type Тип устройств, для которых применяется данное правило политики BYOD. Действительно только для правил BYOD. Синтаксис: DeviceType ::= device_type '(' word | string | list ')' approving_required Подтверждение администратора. Действительно только для правил BYOD. Синтаксис: ApprovingRequired ::= approving_required '(' boolean ')' boolean ::= yes | no | true | false Свойства инспектирования SSH block_ssh_shell Блокировка удалённого запуска shell. Действительно только для правил инспектирования SSH. Синтаксис: BlockSshShell ::= block_ssh_shell '(' boolean ')'

boolean ::= yes | no | true | false block_ssh_exec Блокировка удалённого выполнения команд по SSH. Действительно только для правил инспектирования SSH. Синтаксис: BlockSshExec ::= block_ssh_exec '(' boolean ')' boolean ::= yes | no | true | false block_sftp Блокировка соединения SFTP (Secure File Transfer Protocol). Действительно только для правил инспектирования SSH. Синтаксис: BlockSftp ::= block_sftp '(' boolean ')' boolean ::= yes | no | true | false ssh_command Команда linux, которую требуется передать, в формате ssh user@host 'command'. Действительно только для правил инспектирования SSH. Синтаксис: SshCommand ::= ssh_command '(' string ')' Свойства правил VPN-сервера auth_profile Профиль аутентификации. Действительно только для правил VPN-сервера. Синтаксис: AuthProfile ::= auth_profile '(' string | word ')' vpn_network Сеть VPN. Действительно только для правил VPN-сервера. Синтаксис: VpnNetwork ::= vpn_network '(' string ')' interface Интерфейс VPN. Действительно только для правил VPN-сервера и VPN-клиента. Синтаксис: Interface ::= interface '(' word ')' Свойства правил VPN-клиента server_address IP-адрес сервера. Действительно только для правил VPN-клиента. Синтаксис: ServerAddress ::= server_address '(' ip_address ')' password Пароль. Действительно только для правил VPN-клиента. Синтаксис: Password ::= password '(' word ')' last_error Последняя ошибка VPN. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента. Синтаксис: LastError ::= last_error '(' string ')' connection_time Время соединения. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента. Синтаксис: ConnectionTime ::= connection_time '(' word ')' status Статус. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента. Синтаксис: Status ::= status '(' string ')' Свойства правил сценариев trigger Тип срабатывания сценария. Действительно только для правил сценариев. Синтаксис: Trigger ::= trigger '(' trigger_type ')' trigger_type ::= all_users | one_user duration Продолжительность. Время в минутах, в течение которого сценарий будет активным после его активации. Действительно только для правил сценариев. Синтаксис: Duration ::= duration '(' integer ')' operation_mode Сценарий срабатывает при совпадении одного или всех условий. Синтаксис: OperationMode ::= operation_mode '(' mode ')' mode ::= all | any Свойства условий сценария scond_type Тип условий сценария. Действительно только для условий сценария. Синтаксис: SCondType ::= scond_type '(' type ')' SCondType ::= scond_type '(' type ')' count_interval Количество срабатываний. Действительно только для условий сценария "Категория URL" (url_category), "Приложение" (app), "Тип контента" (mime_type) и "Проверка состояния" (health_check). Синтаксис: CountInterval ::= count_interval '(' integer ')' max_event_count Интервал времени в минутах, за который было обнаружено count_interval срабатываний. Действительно только для условий сценария "Категория URL" (url_category), "Приложение" (app), "Тип контента" (mime_type) и "Проверка состояния" (health_check). Синтаксис: MaxEventCount ::= max_event_count '(' integer ')' packet_size Размер пакета. Размер пакета в трафике пользователя превысил указанное значение. Действительно только для условий сценария "Размер пакета" (net_packet_size). Синтаксис: PacketSize ::= packet_size '(' size ')' size ::= integer | integer KB | integer MB | integer GB 64 — размер в байтах; 2MB — размер в мегабайтах. traffic_limit Объем трафика. Действительно только для условий сценария "Ограничение трафика" (traffic). Синтаксис: TrafficLimit ::= traffic_limit '(' size ')' size ::= integer | integer KB | integer MB | integer GB period Период времени. Действительно только для условий сценария "Ограничение трафика" (traffic). Синтаксис: Period ::= period '(' time_period ')' time_period ::= minute | hour | day | week | month ips_tl Уровень угрозы COB. Действительно только для условий сценария "COB" (ips). Синтаксис: 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 Метод проверки. Действительно только для условий сценария "Проверка состояния" (health_check). Синтаксис: CheckMethod ::= health_check_method '(' method ')' method ::= ping | dns | get health_result Результат. Действительно только для условий сценария "Проверка состояния" (health_check). Синтаксис: esult ::= health_result '(' result ')' result ::=

positive | negative health_request_timeout Тайм-аут подключения (сек). Действительно только для условий сценария "Проверка состояния" (health_check). Синтаксис: RequestTimeout ::= health_request_timeout '(' integer ')' health_answer_timeout Тайм-аут ответа (сек). Действительно только для условий сценария "Проверка состояния" (health_check), "Метод проверки HTTP GET" (get). Синтаксис: ut ::= health_answer_timeout '(' integer ')' health_type_request Тип DNS-запроса. Действительно только для условий сценария "Проверка состояния" (health_check), "метод проверки DNS" (dns). Синтаксис: TypeRequest ::= health_type_request '(' type_name ')' type_name ::= word | string Свойства правил балансировки нагрузки scheduler Метод балансировки. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: 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 Реальные серверы. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: RealServer ::= real_server '(' mode, ipv4[:port], weight ')' mode ::= gate | masq | masq_snat port ::= integer weight ::= integer gate — шлюз; masq — маскарадинг; masq_snat — маскарадинг с подменой IP-адреса источника (SNAT). Пример: 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 Аварийный режим сервера. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: IpvsFallback ::= ipvs_fallback '(' mode, ipv4[:port] ')' mode ::= gate | masq | masq_snat port ::= integer monitor_kind Режим. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorKind ::= monitor_kind '(' kind ')' kind ::= ping | connect | negotiate monitor_service Сервис. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorService ::= monitor_service '(' service ')' service ::= http | dns monitor_request Запрос. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorRequest ::= monitor_request '(' string ')' monitor_response Ожидаемый ответ. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorResponse ::= monitor_response '(' string ')' monitor_interval Интервал проверки. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorInterval ::= monitor_interval '(' integer ')' monitor_timeout Время ожидания. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorTimeout ::= monitor_timeout '(' integer ')' monitor_failurecount Число неудачных попыток. Действительно только для правил балансировки нагрузки TCP/UDP. Синтаксис: MonitorFailurecount ::= monitor_failurecount '(' integer ')'

28

Действия

33

Типы правил

38

Список поддерживаемых HTTP-заголовков

51

Список категорий

53

Product: DCFW

Version: 8.x

UPL (UserGate Policy Language) — язык описания политик UserGate. Термин «политика» употребляется здесь в контексте конфигурации правил, применяемых для принятия решений по требованиям аутентификации, правам доступа или преобразования контента.

Общие положения языка UPL

Правила настраиваются с использованием действий, условий и свойств.

Для каждого правила настраивается одно из действий. Действия — настройки, которые управляют обработкой транзакции (OK, WARNING, PASS, DENY, FORCE_PASS, FORCE_DENY). При настройке правил, в которых не предусмотрено указание действия (например, правила DNS, NAT и маршрутизации, пропускной способности и т.п.), необходимо указать действия PASS или OK.

Условия задаются знаками равно (=) или не равно (!=), например, зоны, адреса, GeoIP источников и назначения, сервисы, приложения и т.д.; все условия в правиле проверяются по логическому И, т.е. правило сработает, если будут выполнены все условия.

Свойства правил задаются в круглых скобках и используются для указания дополнительной информации, например, название правил, их описание, функция журналирования и т.д.

Примечание При настройке правил сначала указывается действие, потом условия и затем свойства.

UPL используется для помощи UPL в интерфейсе CLI создаются правила политик сети и безопасности для следующих разделов:

- Настройки DNS-прокси (уровень: **network dns dns-proxy dns-rules**).
- Captive-портал (уровень: **users captive-portal**).
- Межсетевой экран (уровень: **network-policy firewall**).
- NAT и маршрутизация (уровень: **network-policy nat-routing**).
- Пропускная способность (уровень: **network-policy traffic-shaping**).
- Фильтрация контента (уровень: **security-policy content-filtering**).
- Веб-безопасность (уровень: **security-policy safe-browsing**).
- Инспектирование туннелей (уровень: **security-policy tunnel-inspection**).
- Инспектирование SSL (уровень: **security-policy ssl-inspection**).
- Инспектирование SSH (уровень: **security-policy ssh-inspection**).
- COB (уровень: **security-policy intrusion-prevention**).
- Защита почтового трафика (уровень: **security-policy mail-security**).
- ICAP-правила (уровень: **security-policy icap-rules**).
- Правила защиты DoS (уровень: **security-policy dos-rules**).
- Веб-портал (уровень: **global-portal web-portal**).
- Правила reverse-прокси (уровень: **global-portal reverse-proxy-rules**).
- Серверные правила VPN (уровень: **vpn server-rules**).
- Клиентские правила VPN (уровень: **vpn client-rules**).

Структура команды для создания правила:

```
Admin@nodename# create <level> <position> upl-rule <str-upl-syntax>
```

где <level> — уровень, на котором необходимо создать правило.

<position> — позиция, на которую будет помещено правило.

<str-upl-syntax> строка, в которой описано правило в UPL синтаксисе.

Структура команды для обновления существующего правила:

```
Admin@nodename# set <level> <position> upl-rule <str-upl-syntax>
```

где <level> — уровень, на котором необходимо обновить правило.

<position> — номер правила, которое необходимо обновить.

<str-upl-syntax> строка, в которой описано правило в UPL синтаксисе.

Структура команды для удаления правила:

```
Admin@nodename# delete <level> <position> | all>
```

где <level> — уровень, на котором необходимо удалить правило.

<position> — номер правила, которое необходимо удалить.

<all> — удалить все правила.

Структура команды для отображения правила:

```
Admin@nodename# show <level> <position> | all>
```

где <level> — раздел, правила которого нужно отобразить.

<position> — номер правила, которое необходимо отобразить.

<all> — отобразить все правила.

Пример создания правила межсетевого экрана с использованием UPL (использован многострочный ввод):

```
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)
```

После создания правило отобразится в начале списка правил межсетевого экрана (на позиции 1). Данное правило запрещает HTTPS-трафик из зоны "Trusted" в зону "Untrusted" пользователям, идентифицированным системой. Правило работает в соответствии с расписанием "Working hours". При срабатывании правила в журнал будет записана информация о начале сессии.

Комментарии

Любая строка, начинающаяся с символа "%", является комментарием.

Символ процента "%" после пробела или табуляции вводит комментарий, который продолжается до конца строки (кроме случаев, когда символ процента отображается внутри кавычек ("")), как часть выражения).

Пример:

```
% Это комментарий
DENY("Too many Host headers") request.header.Host.count = 2.. % и это тоже
```

Комментарии могут быть в любом месте файла с описанием политик.

Правила

Правило политики (rule) состоит из условий и некоторого количества действий, записанных в любом порядке. Есть также свойства (properties), которые синтаксически выглядят как действие, но при этом активных действий не производят. Например, свойство *late* просто добавляет атрибут "имя" в правило.

Правила обычно пишутся в одной строке, но могут быть разбиты на строки с помощью специального символа — обратного слеша " \".

Когда правило выполняется, условие проверяется для текущей конкретной транзакции. Если условие оценивается как *True* (истина), выполняются все перечисленные действия и текущий слой заканчивается при наличии префиксов *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*. Если сработавшее правило не имеет префиксов *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*, то выполняются действия и дальше обрабатывается уже следующее правило. Если условие оценивается как *False* для этой транзакции, то дальше обрабатывается уже следующее правило.

Все условия в правиле проверяются по логическому "И". Другими словами, правило сработает, когда будут выполнены все условия.

В свою очередь, условие является логической комбинацией триггеров. Триггеры — это отдельные тесты, которые можно выполнить с компонентами запроса, ответа, связанными пользователями или состоянием системы.

Действия — это настройки, которые управляют обработкой транзакции. Например, запретить (*deny*) или обработать объект (изменить заголовок — *rewrite*).

Синтаксис:

```
Rule ::= (PASS | FORCE_PASS | DENY | ( DENY (' string ') ) | FORCE_DENY | FORCE_DENY (' string ') | WARNING | OK)? Conditions '?' Actions
```

```
Conditions ::= condition '?' Conditions
```

```
Actions ::= action '?' Actions
```

Пример:

Запрос будет запрещен, когда сработают оба триггера:

- домен будет example.com
- время будет между 9 и 17 часами

```
DENY url.domain = "example.com" time=09:00..17:00
```

Слой

Слой (layer)— это конструкция UPL, используемая для группировки правил и принятия одного решения. Раздельные принятия решения помогают контролировать сложность политики. Это делается путем написания каждого решения в отдельном слое.

У любого правила в слое может быть префикс *PASS / FORCE_PASS / DENY / FORCE_DENY / OK / WARNING*, когда срабатывает правило с таким префиксом, все остальные правила в слое пропускаются.

В случае если сработало правило с префиксом *FORCE_PASS* или *FORCE_DENY*, то это является окончательным результатом обработки, в противном случае обработка переходит на следующий слой. После обработки всех слоев запрос будет заблокирован или пропущен в зависимости от того, что было последним — *PASS / FORCE_PASS* или *DENY / FORCE_DENY*. Если процессинг остановится на *WARNING*, будет добавлено предупреждение в тело ответа.

Префикс *OK* подразумевает остановку обработки правил в текущем слое при выполнении условий и действий (если таковые указаны). Если префикс отсутствует при выполнении условий и действий, то остановка не подразумевается.

Действия *FORCE_PASS* и *FORCE_DENY* похожи на *PASS* и *DENY*, за исключением того, что они могут быть переопределены на последующих слоях. *FORCE_DENY* и *FORCE_PASS* немедленно прекращают проверку правил как на текущем, так и на последующих слоях, и этот результат является окончательным.

Синтаксис:

```
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 ::= "" произвольная строка ""

Пример 1:

```
[content "L1"]
DENY enabled(true) % по умолчанию все запрещено
```

```
[content "Devs"]
DENY group != Developers enabled(true)
%... дальше идут правила, которые будут применяться только для группы Developers
```

Пример 2:

```
[content "Admin"]
FORCE_PASS group = Admins enabled(true)
```

```
[content "L2"]
DENY enabled(true) % по умолчанию все запрещено
```

Динамические значения

Значения "адрес запроса" (*url, url.host, url.path*), "IP-адрес источника/назначения" (*src.ip, dst.ip*), "имя пользователя" (*user*), "значения заголовков" (*request* и *response*) и "параметры запроса" (*qparam*) могут сравниваться между собой, а также использоваться в качестве аргумента в действиях (*actions*), где это предусмотрено.

Условия

Version: 8.x
Product: DCFW

Условие (condition) в языке UPL является логической комбинацией триггеров. Триггеры — это отдельные тесты, которые можно выполнить с компонентами запроса, ответа, связанными пользователями или состоянием системы. Все триггеры условия сравниваются со значениями с помощью операторов "=" и "!=". В роли значения могут выступать константные значения, такие как строки, целочисленные значения, диапазоны значений, динамические значения.

Синтаксис:

```
condition ::= condition_name ('=' | '!=') condition_value  
condition_value ::= pattern | list  
list ::= '(' ((pattern ',' )* pattern)? ')'  
pattern ::= word | string | integer | float | boolean | range | condition_name  
string ::= "" произвольная строка ""  
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

Проверка наличия *HTTP CONNECT*.

Синтаксис:

```
http.connect = yes | no | true | false
```

http.method

Проверка используемого HTTP-метода. Метод можно указывать как в кавычках, так и без.

Синтаксис:

```
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

Проверка версии HTTP-запроса.

Синтаксис:

```
http.request.version = 0.9 | 1.0 | 1.1
```

http.response.version

Проверка версии HTTP-ответа.

Синтаксис:

```
http.response.version = 0.9 | 1.0 | 1.1
```

http.response.code

Проверка HTTP-кода ответа. Валидные значения: 100 - 999.

Синтаксис:

```
http.response.code = NNN %(где NNN число от 100 до 999)
```

http.request.body, http.request.body.nocase, http.response.body и http.response.body.nocase

Проверка тела запроса/ответа HTTP на содержание определенной сигнатуры.

Пример:

```
DENY http.response.body.nocase = "<title>index of" http.response.body = ">"
```

category

Проверка домена на принадлежность к определенной категории сайтов (смотрите в Приложении [список категорий](#)). Категорию можно указать как по имени, так и по ее идентификатору. Если имя категории содержит пробелы или спецсимволы, ее необходимо указывать в кавычках.

Синтаксис:

```
category = word | string | integer | list | lib
```

```
lib ::= lib.category '(' list_libs ')'
```

```
list_libs ::= lib_name ',' list_libs
```

```
lib_name ::= word | string
```

Пример:

Запретить категории *Job Search* и *Gambling*:

```
DENY category = ("Job Search", Gambling)
```

Запретить все категории из библиотеки *Restricted cats*:

```
DENY category = lib.category("Restricted cats")
```

morphology

Проверка тела ответа морфологическими словарями.

Синтаксис:

```
morphology = word | string | list | lib
```

```
lib ::= lib.morphology '(' list_libs ')'
```

```
list_libs ::= lib_name ',' list_libs
```

```
lib_name ::= word | string
```

Пример:

Запретить контент, если сработает категория морфологии из словаря *BadWords*:

```
DENY morphology = BadWords
```

Запретить морфологические категории из библиотек **Special Words**, **BadWords**

```
DENY morphology = ("Special Words", BadWords)
```

```
DENY morphology = lib.morphology("Special Words", BadWords) % аналогично предыдущему правилу
```

request.header.<h_name> и response.header.<h_name>

Проверка HTTP-заголовка запроса/ответа. *h_name* может принимать одно из поддерживаемых значений (смотрите список поддерживаемых HTTP-заголовков в [Приложении](#)).

Синтаксис:

```
request.header.<h_name>[.base64][.nocase] = string
```

Пример:

```
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 и response.header.<h_name>.substring

Проверка HTTP-заголовка запроса/ответа на вхождение подстроки. *h_name* может принимать одно из поддерживаемых значений (смотрите список поддерживаемых HTTP-заголовков в [Приложении](#)).

Синтаксис:

```
request.header.<h_name>[.base64]substring[.nocase] = string
```

Пример:

```
DENY request.header.User-Agent.substring = "curl/"
```

request.header.<h_name>.regex и response.header.<h_name>.regex

Проверка HTTP-заголовка запроса/ответа на регулярное выражение PCRE. *h_name* может принимать одно из поддерживаемых значений (смотрите список поддерживаемых HTTP-заголовков в [Приложении](#)).

Синтаксис:

```
request.header.<h_name>[.base64].regex = string
```

Пример:

```
DENY("Accept only digits in content length") request.header.Content-Length.regex != "[0-9]*"
```

request.header.<h_name>.re2 и response.header.<h_name>.re2

Проверка HTTP-заголовка запроса/ответа на регулярное выражение RE2. *h_name* может принимать одно из поддерживаемых значений (смотрите список поддерживаемых HTTP-заголовков в [Приложении](#)).

Синтаксис:

```
request.header.<h_name>[.base64].re2 = string
```

Пример:

DENY("Accept only digits in content length") request.header.Content-Length.re3 != "[0-9]"

request.header.<h_name>.count и response.header.<h_name>.count

Проверка количества заголовков <h_name> в HTTP-запросе/ответе. *h_name* может принимать одно из поддерживаемых значений (смотрите список поддерживаемых HTTP-заголовков в [Приложении](#)).

Синтаксис:

request.header.<h_name>.count = integer | range

Пример:

DENY("Too many Host headers") request.header.Host.count = 2..

request.header.<h_name>.length и response.header.<h_name>.length

Проверка длины значений всех заголовков <h_name> в HTTP-запросе/ответе. *h_name* может принимать одно из поддерживаемых значений (смотрите список поддерживаемых HTTP-заголовков в [Приложении](#)).

Синтаксис:

request.header.<h_name>.length = integer | range

Пример:

DENY("Too much Cookie data") request.header.Cookie.length = 2048..

request.header_names, request.header_values, response.header_values и response.header_values

Проверка имени/значения всех HTTP-заголовков запроса/ответа на значение.

Синтаксис:

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> и response.x_header.<xh_name>

Проверка HTTP-заголовка запроса/ответа на значение. *xh_name* — произвольный HTTP-заголовок.

Синтаксис:

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

Пример:

DENY url="http://usergate.com" request.x_header.Test="test1"

Возможны также суффиксы **length**, **count**, **regex**, **re2** как и в случае с <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>

Проверка заголовка запроса Cookie на значение.

Синтаксис:

request.header.Cookie.<cookie_name>[.base64][.nocase | substring | substring.nocase | regex | re2] = string

Пример:

DENY http.method = POST request.header.Cookie.csrf_token != qparam.CSRF_TOKEN enabled(true) name("Check CSRF")

time, day, hour, minute

Проверка соответствия текущего времени заданному условию. Если не указан суффикс *utc*, время берется локальное, иначе — по Гринвичу.

Синтаксис:

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

Пример:

PASS time = 12:00..13:00 % разрешить каждый день с 12 до 13 часов

PASS time = lib.time("Праздники") % использовать библиотеку "Праздники"

DENY day = (sunday, saturday) % запретить на выходных

DENY day = (monday, 15) hour = 9..18 % запретить каждый понедельник и каждое 15 число месяца с 9 до 18 часов

Открытые интервалы учитываются по границе суток/часа.

PASS hour = 18.. % означает, что разрешено с 18 часов до полуночи

minute = ..10 % первые 10 минут каждого часа

url, url.host и url.address

Проверка url или его части на значение. Проверка использует нормализованный URI с декодированными * %*.

Синтаксис:

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 % полный или нет URL

prefix ::= string % начало строки

substring ::= string % подстрока

suffix ::= string % окончание строки

regex ::= string % регулярное выражение PCRE

re2 ::= string % регулярное выражение RE2

url.address — это, по сути, синоним *dst.ip*.

Пример:

DENY url.path.base64.re2 = "(?i)bondisconnecting\W*" enabled(true) name("ondisconnecting (URI)")

qparam.<name>, qparam.values и qparam.names

Проверка значения параметров запроса. Проверка использует имена и значения параметров с декодированными * %*.

Синтаксис:

qparam.length = numeric % проверить общую длину query-параметров

qparam.count = numeric % проверить количество query-параметров *qparam.<name>[(.length | count)] = numeric*

qparam.<name>[.base64][(.nocase | substring | substring.nocase | regex | re2)] = string

qparam.values[.base64].substring[.nocase] = string % проверить все значения на вхождение подстроки

qparam.names[.base64].substring[.nocase] = string % проверить все имена на вхождение подстроки

qparam.values[.base64].regex = string % проверить все значения на регулярное выражение

qparam.names[.base64].regex = string % проверить все имена на регулярные выражения

qparam.values[.base64].re2 = string % проверить все значения на регулярное выражение

qparam.names[.base64].re2 = string % проверить все имена на регулярные выражения *numeric ::= integer | range* % число либо диапазон

regex ::= string % регулярное выражение PCRE

re2 ::= string % регулярное выражение RE2

Пример:

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 и group

Проверка текущего пользователя или его группы.

Синтаксис:

user = word | string | known | unknown

group = word | string

user.guid = string

group.guid = string

known — используется для указания авторизованного (известного) пользователя;

unknown — используется для указания неавторизованного (неизвестного) пользователя.

Пример:

PASS user = known % разрешить известных пользователей

DENY group = "Отдел продаж" category = "Pornography/Sexually Explicit" % запретить категорию порно для группы "Отдел продаж"

src и dst

Проверка условия на IP-адрес, зону или GeoIP источника/назначения.

Синтаксис:

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 — это, по сути, синоним *dst.ip*.

scenario

Проверка активности определенного сценария.

Синтаксис:

scenario = string | word | list

Пример:

DENY scenario = Torrents desc("Запретить, если активен сценарий Torrents")

virus_heuristic и virus_usergate

Проверка тела ответа на вирусы.

Синтаксис:

virus_heuristic = yes | no | true | false % (по умолчанию false)

virus_usergate = yes | no | true | false % (по умолчанию false)

heuristic — эвристический анализатор (медленный);

usergate — проверка по хэшам (быстрый).

bridge_vlan_filter

Фильтрация трафика по VLAN-тегам для интерфейса в режиме моста (bridge).

Синтаксис:

bridge_vlan_filter '='|'!=' list | number | number..number

Пример:

DENY bridge_vlan_filter = (10, 100..200) desc("Запретить трафик по VLAN-терам")

service

Детектирование трафика определенного сервиса для межсетевого экрана.

Синтаксис:

service = string | word | list

Пример:

DENY service = POP3 desc("Запретить сервис POP3")

application

Детектирование трафика определенного приложения на уровне L7 для межсетевого экрана.

Синтаксис:

application = string | word | list

Пример:

DENY application = Tor desc("Запретить Tor")

envelope_from и envelope_to

Проверка email-адреса отправителя/адресата сообщения.

Синтаксис:

envelope_from '='|'!=' string | list

envelope_to '='|'!=' string | list

Пример:

PASS envelope_from = "Email froup from" envelope_to = "Email froup to" service = SMTP mark_hdr(Subject) enabled(true) name("Mail Pass Rule")

response_time

Проверка времени ответа в миллисекундах.

Синтаксис:

response_time = integer

hip_profile

HIP-профили для проверки соответствия конечного устройства требованиям безопасности. Действительно только для правил межсетевого экрана.

Синтаксис:

hip_profile = string | word | list

Встроенные библиотеки

Version: 8.x

Product: DCFW

Библиотеки (*lib*) — это элементы языка UPL, которые служат для доступа к встроенным и пользовательским библиотекам. Как правило, это достаточно большие списки, которые неудобно описывать через определения **def**. Обращение к библиотекам происходит по их именам.

Синтаксис:

library ::= *lib*.<*url* | *morphology* | *category* | *useragent* | *mime* | *network* | *time* | *applicationgroup* | *servicegroup*>(*list_names*)

list_names ::= *name list_names*

name ::= *word* | *string*

url — список URL;

morphology — список морфологических словарей;

category — группа категорий;

useragent — список юзерагентов;

mime — список типов контента;

network — список сетей/IP-адресов;

time — библиотека с промежутками времени.

Пример:

DENY src.ip = lib.network("Bad ips", "Test ips")

DENY dst.ip = lib.network("Bad ips")

DENY dst.ip = lib.url("Bad urls") % в данном случае домены будут резолвиться в ip-адреса

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)

Определения

Version: 8.x

Product: DCFW

В файлах политик определения (**def**) служат для объединения наборов условий или действий. Каждое определение должно иметь уникальное пользовательское имя, по которому к нему можно обратиться из правил.

def condition

Наборы условий. Все условия в одной строке проверяются по логическому *И*. Перевод строки означает логическое *ИЛИ*. Символ экранирования — обратный слэш ("\\") в конце строки позволяет перенести условие по **И** на следующую строку.

Синтаксис:

```
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

Список условий сценария. Каждое условие сценария обычно пишется в одной строке, но при необходимости условие может быть разбито на строки с помощью специального символа — обратного слеша ("\\").

Синтаксис:

```
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

Определение переменных. Служит для подсчета некоторых событий за определенный интервал времени. Для изменения значения предназначены действия **inc** и **dec**.

Синтаксис:

```
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 — это начальное значение переменной, к которому она вернется по истечении времени **window**;

key — поле или список полей, по которым группируются значения переменной (необязательный параметр).

Свойства

Version: 8.x
Product: DCFW

Свойства (properties) — это некие атрибуты правила, например, *name* или *enabled*. Они используются для предоставления дополнительной информации в процессе обработки правил. Синтаксис свойств точно такой же, как у действий.

Синтаксис:

```
property = prop_name | prop_name '(' list_params ')'
prop_name ::= name | desc | id | rule_log | enabled | scenario
list_params ::= value ',' list_params
```

name и desc

Атрибуты *имя* и *описание* для правила.

Синтаксис:

```
Name ::= name '(' string|word ')'
Description ::= desc '(' string ')'
```

Пример:

DENY hour = 9..18 category = News name("Запретить News") desc("Запретить категорию News в рабочее время")

enabled

Атрибут, который включает или выключает работу правила.

Синтаксис:

```
Enable ::= enabled '(' boolean ')'
boolean ::= yes | no | true | false      % (по умолчанию false)
```

rule_log

Устанавливает атрибут журналирования правила. Значение *session* действительно только для правил межсетевого экрана, защиты от dos-атак и пропускной способности.

Синтаксис:

```
Logging ::= rule_log '(' boolean | session ')'
LoggingFwRule ::= rule_log '(' boolean , interval, burst')'
boolean ::= yes | no | true | false      % (по умолчанию no)
interval ::= "integer/[s,m,h,d]"
burst ::= integer
```

interval — среднее число пакетов, попадающих под условие *limit* в единицу времени (1/s, 1/m, 1/h, 1/d) , default = 3/h;
burst — максимальное число пакетов, попадающих в под условие *limit* за один раз (default = 5).

profile

Устанавливает профиль правила.

Синтаксис:

```
Profile ::= profile '(' string | word | list ')'
```

certificate

Сертификат, используемый для поддержки HTTPS-соединения. Действительно только для правил reverse-прокси.

Синтаксис:

```
CertAuthEnabled ::= cert_auth_enabled '(' boolean ')'
Certificate ::= certificate '(' certificate_name ')'
certificate_name ::= string | word
```

gateway

Шлюз. Имя одного из существующих шлюзов. Действительно только для правил NAT и маршрутизации, и для условий сценария "Проверка состояния".

Синтаксис:

```
Gateway ::= gateway '(' string | word ')'
```

Свойства правил межсетевого экрана

reject_with

Устанавливает способ, с помощью которого будет блокироваться трафик. Действительно только для правил межсетевого экрана.

Синтаксис:

```
Reject ::= reject_with '(' "tcp-reset-both" | "tcp-rst" | "host-unreach" )'
```

fragmented

Проверка на фрагментированность пакетов. Действительно только для правил межсетевого экрана.

Синтаксис:

```
Fragmented ::= fragmented '(' boolean ')'
```

```
boolean ::= yes | no | true | false
```

yes — проверяются только фрагментированные пакеты;

no — проверяются только нефрагментированные пакеты;

—, если свойство *fragmented* не указано, то будут проверяться все пакеты.

ips_profile

Устанавливает профиль IPS. Действительно только для правил межсетевого экрана.

Синтаксис:

```
IPS_Profile ::= ips_profile '(' string | word ')'
```

L7_profile

Устанавливает профиль приложения. Действительно только для правил межсетевого экрана.

Синтаксис:

```
L7_Profile ::= L7_profile '(' string | word ')'
```

Свойства правил инспектирования SSL

block_invalid_cert

Блокирование сайтов с некорректными сертификатами. Действительно только для правил инспектирования SSL.

Синтаксис:

```
InvalidCertificate ::= block_invalid_cert '(' boolean ')'
```

```
boolean ::= yes | no | true | false
```

check_revoc_cert

Проверка по списку отозванных сертификатов. Действительно только для правил инспектирования SSL.

Синтаксис:

```
ChekRevocation ::= check_revoc_cert '(' boolean ')'
```

```
boolean ::= yes | no | true | false
```

block_expired_cert

Блокировка сертификатов с истекшим сроком действия. Действительно только для правил инспектирования SSL.

Синтаксис:

```
ExpiredCertificate ::= block_expired_cert '(' boolean ')'
```

```
boolean ::= yes | no | true | false
```

block_self_signed_cert

Блокировка самоподписанных сертификатов. Действительно только для правил инспектирования SSL.

Синтаксис:

```
SelfSignedCertificate ::= block_self_signed_cert '(' boolean ')'
```

boolean ::= yes | no | true | false

ssl_profile

Профиль SSL. Действительно только для правил инспектирования SSL, reverse-прокси, веб-портала.

Синтаксис:

SslProfile ::= ssl_profile '(' string | word ')'

ssl_forward_profile

Профиль пересылки SSL. Действительно только для правил инспектирования SSL.

Синтаксис:

SslForwardProfile ::= ssl_forward_profile '(' string | word ')'

profile_id ::= integer

Свойства правил пропускной способности

bandwidth_pool

Полоса пропускания. Действительно только для правил пропускной способности.

Синтаксис:

BandwidthPool ::= bandwidth_pool '(' bandwidth ')'

bandwidth ::= "100 Kbps" | "512 kbps" | "1 Mbps" | "2 Mbps" | "5 Mbps" | "10 Mbps" | "20 Mbps" | "50 Mbps" | "100 Mbps"

Пример:

scenario = "Сценарий обнаружения torrent" bandwidth_pool("1 Mbps") enable(true) name("Пропускная способность torrent")

Свойства правил защиты почтового трафика

mark

Маркировка. Текст тега, который маркирует письмо. Действительно только для правил защиты почтового трафика.

Синтаксис:

Mark ::= mark '(' string | word ')'

mark_hdr

Заголовок. Поле, куда помещается тег маркировки. Действительно только для правил защиты почтового трафика.

Синтаксис:

MarkHeader ::= mark_hdr '(' word ')'

antispam_kav

Проверка антиспамом. Действительно только для правил защиты почтового трафика.

Синтаксис:

AntispamKav ::= antispam_kav '(' boolean ')'

boolean ::= yes | no | true | false

antispam_usergate

Проверка антиспамом UserGate. Действительно только для правил защиты почтового трафика.

Синтаксис:

AntispamUsergate ::= antispam_usergate '(' boolean ')'

boolean ::= yes | no | true | false

Пример:

DENY("with error") envelop_to = UserGate antispam_usergate(yes)

dnsbl

DNSBL-проверка (только SMTP). Действительно только для правил защиты почтового трафика.

Синтаксис:

DNSBLacklistCheck ::= dnsbl '(' boolean ')'

boolean ::= yes | no | true | false

Пример:

DENY service = SMTP dnsbl(yes)

Свойства правил NAT и маршрутизации

target_ip

Адрес назначения DNAT, если задано действие *dnat* или *port_mapping*.

Новая IP-сеть/маска, если задано действие *netmap*.

Действительно только для правил NAT и маршрутизации.

Синтаксис:

TargetIpl ::= target_ip '(' ipv4 | ipv4_with_mask ')'

Пример:

PASS target_ip("192.168.1.20") dnat

target_snat

Включение SNAT. При включении данной опции UserGate будет изменять адрес источника в пакетах из внешней сети на свой IP-адрес.

Действительно только для правил NAT и маршрутизации.

Синтаксис:

TargetSnat ::= target_snat '(' boolean ')'

boolean ::= yes | no | true | false

snat_target_ip

SNAT IP (внешний адрес). Явно указывает IP-адрес, на который будет заменен адрес источника при подмене адресов пакетов.

Действительно только для правил NAT и маршрутизации.

Синтаксис:

SnatTargetIpl ::= snat_target_ip '(' ip_address ')'

port_map

Порт-форвардинг. Переопределение портов публикуемых сервисов. Действительно только для правил NAT и маршрутизации.

Синтаксис:

PortMap ::= port_map '(' protocol, port_from, port_to ')'

protocol ::= tcp | udp | smtp | smpts

port_from ::= integer

port_to ::= integer

port_from — номер TCP/UDP-порта, на который пользователи шлют запросы;

port_to — номер TCP/UDP-порта, на который будут пересылаться запросы пользователей на внутренний публикуемый сервер.

direction

Направление подмены сетей. Действительно только для правил NAT и маршрутизации.

Синтаксис:

Direction ::= direction '(' input | output ')'

input — входящий, подменяется IP-сеть назначения. Будут изменены IP-адреса назначения в трафике;

output — исходящий, подменяется IP-сеть источника. Будут изменены IP-адреса источника в трафике.

Свойства правил DNS-прокси

dns_server

Список IP-адресов DNS-серверов. Действительно только для правил DNS-прокси.

Синтаксис:

DnsServer ::= dns_server '(' ip_address | ip_address_list ')'

Свойства правил reverse-прокси

cert_auth_enabled

Аутентификация по сертификату. Действительно только для правил Reverse-прокси.

Синтаксис:

CertAuthEnabled ::= cert_auth_enabled '(' boolean ')'

boolean ::= yes | no | true | false

is_https

Включение поддержки HTTPS. Действительно только для правил Reverse-прокси.

Синтаксис:

IsHttps ::= is_https '(' boolean ')'

boolean ::= yes | no | true | false

rewrite_path

Подмена путей. Действительно только для правил Reverse-прокси.

Синтаксис:

RewritePath ::= rewrite_path '(' path_from, path_to ')'

path_from — изменить с (домен и/или путь URL, которые требуется изменить);

path_to — изменить на (домен и/или путь URL, на которые требуется заменить старые).

waf_profile

Профиль WAF. Действительно только для правил Reverse-прокси.

Синтаксис:

WafProfile ::= waf_profile '(' string | word | list ')'

Свойства правил веб-портала

icon

Иконка, которая будет отображаться на веб-портале для данной закладки. Действительно только для правил веб-портала.

Синтаксис:

Icon ::= icon '(' string | word ')'

additional_url

Вспомогательные URL, необходимые для работы основного URL, но которые нет необходимости публиковать для пользователей.

Действительно только для правил веб-портала.

Синтаксис:

AdditionalUrl ::= additional_url '(' string | word | list ')'

rdp_check_session_alive

Проверка авторизации для RDP-сессий. Действительно только для правил веб-портала.

Синтаксис:

RdpCheckSessionAlive ::= rdp_check_session_alive '(' boolean ')'

boolean ::= yes | no | true | false

transparent_auth

Включение прозрачной аутентификации пользователя. Действительно только для правил веб-портала.

Синтаксис:

TransparentAuth ::= transparent_auth '(' boolean ')'

boolean ::= yes | no | true | false

Свойства правил веб-безопасности (Safe browsing)

enable_adblock

Блокировка рекламы.

Синтаксис:

EnableAdblock ::= enable_adblock '(' boolean ')'

boolean ::= yes | no | true | false

url_list_exclusions

URL-список сайтов исключений, для которых блокировать рекламу не требуется.

Синтаксис:

UrlListExclusions ::= url_list_exclusions '(' list_libs ')'

list_libs ::= lib_name ';' list_libs

lib_name ::= word | string

enable_injector

Позволяет вставить произвольный код во все веб-страницы.

Синтаксис:

EnableInjector ::= enable_injector '(' boolean ')'

boolean ::= yes | no | true | false

custom_injector

Код инжектора.

Синтаксис:

CustomInjector ::= custom_injector '(' string ')'

safe_search

Функция безопасного поиска.

Синтаксис:

SafeSearch ::= safe_search '(' boolean ')'

boolean ::= yes | no | true | false

search_history_logging

Запись в журнал поисковых запросов пользователей.

Синтаксис:

SearchHistoryLogging ::= search_history_logging '(' boolean ')'

boolean ::= yes | no | true | false

social_sites_block

Блокировка приложения в популярных социальных сетях.

Синтаксис:

SocialSitesBlock ::= social_sites_block '(' boolean ')'

boolean ::= yes | no | true | false

Свойства BYOD

max_device_number

Максимальное количество устройств, с которых пользователь может получать доступ в сеть.
Действительно только для правил BYOD.

Синтаксис:

MaxDeviceNumber ::= max_device_number '(' integer ')'

max_active_device_number

Максимальное количество устройств, с которых пользователь одновременно может получать доступ в сеть.
Действительно только для правил BYOD.

Синтаксис:

MaxActiveDeviceNumber ::= max_active_device_number '(' integer ')'

device_type

Тип устройств, для которых применяется данное правило политики BYOD.
Действительно только для правил BYOD.

Синтаксис:

DeviceType ::= device_type '(' word | string | list ')'

approving_required

Подтверждение администратора.
Действительно только для правил BYOD.

Синтаксис:

ApprovingRequired ::= approving_required '(' boolean ')'

boolean ::= yes | no | true | false

Свойства инспектирования SSH

block_ssh_shell

Блокировка удалённого запуска shell. Действительно только для правил инспектирования SSH.

Синтаксис:

BlockSshShell ::= block_ssh_shell '(' boolean ')'

boolean ::= yes | no | true | false

block_ssh_exec

Блокировка удалённого выполнения команд по SSH. Действительно только для правил инспектирования SSH.

Синтаксис:

BlockSshExec ::= block_ssh_exec '(' boolean ')'

boolean ::= yes | no | true | false

block_sftp

Блокировка соединения SFTP (Secure File Transfer Protocol). Действительно только для правил инспектирования SSH.

Синтаксис:

BlockSftp ::= block_sftp '(' boolean ')'

boolean ::= yes | no | true | false

ssh_command

Команда linux, которую требуется передать, в формате ssh user@host 'command'.

Действительно только для правил инспектирования SSH.

Синтаксис:

SshCommand ::= ssh_command '(' string ')'

Свойства правил VPN-сервера

auth_profile

Профиль аутентификации. Действительно только для правил VPN-сервера.

Синтаксис:

AuthProfile ::= auth_profile '(' string | word ')'

vpn_network

Сеть VPN. Действительно только для правил VPN-сервера.

Синтаксис:

VpnNetwork ::= vpn_network '(' string ')'

interface

Интерфейс VPN. Действительно только для правил VPN-сервера и VPN-клиента.

Синтаксис:

Interface ::= interface '(' word ')'

Свойства правил VPN-клиента

server_address

IP-адрес сервера. Действительно только для правил VPN-клиента.

Синтаксис:

ServerAddress ::= server_address '(' ip_address ')'

password

Пароль. Действительно только для правил VPN-клиента.

Синтаксис:

Password ::= password '(' word ')'

last_error

Последняя ошибка VPN. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента.

Синтаксис:

LastError ::= last_error '(' string ')'

connection_time

Время соединения. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента.

Синтаксис:

ConnectionTime ::= connection_time '(' word ')'

status

Статус. Информационное поле, недоступное для редактирования. Действительно только для правил VPN-клиента.

Синтаксис:

Status ::= status '(' string ')'

Свойства правил сценариев

trigger

Тип срабатывания сценария. Действительно только для правил сценариев.

Синтаксис:

Trigger ::= trigger '(' trigger_type ')'

trigger_type ::= all_users | one_user

duration

Продолжительность. Время в минутах, в течение которого сценарий будет активным после его активации. Действительно только для правил сценариев.

Синтаксис:

Duration ::= duration '(' integer ')'

operation_mode

Сценарий срабатывает при совпадении одного или всех условий.

Синтаксис:

OperationMode ::= operation_mode '(' mode ')'

mode ::= all | any

Свойства условий сценария

scond_type

Тип условий сценария. Действительно только для условий сценария.

Синтаксис:

SCondType ::= scond_type '(' type ')'

SCondType ::= scond_type '(' type ')'

count_interval

Количество срабатываний. Действительно только для условий сценария "Категория URL" (url_category), "Приложение" (app), "Тип контента" (mime_type) и "Проверка состояния" (health_check).

Синтаксис:

CountInterval ::= count_interval '(' integer ')'

max_event_count

Интервал времени в минутах, за который было обнаружено *count_interval* срабатываний. Действительно только для условий сценария "Категория URL" (url_category), "Приложение" (app), "Тип контента" (mime_type) и "Проверка состояния" (health_check).

Синтаксис:

MaxEventCount ::= max_event_count '(' integer ')'

packet_size

Размер пакета. Размер пакета в трафике пользователя превысил указанное значение. Действительно только для условий сценария "Размер пакета" (net_packet_size).

Синтаксис:

PacketSize ::= packet_size '(' size ')'

size ::= integer | integer KB | integer MB | integer GB

64 — размер в байтах;

2MB — размер в мегабайтах.

traffic_limit

Объем трафика. Действительно только для условий сценария "Ограничение трафика" (traffic).

Синтаксис:

TrafficLimit ::= traffic_limit '(' size ')'

size ::= integer | integer KB | integer MB | integer GB

period

Период времени. Действительно только для условий сценария "Ограничение трафика" (traffic).

Синтаксис:

Period ::= period '(' time_period ')'

time_period ::= minute | hour | day | week | month

ips_tl

Уровень угрозы COB. Действительно только для условий сценария "COB" (ips).

Синтаксис:

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

Метод проверки. Действительно только для условий сценария "Проверка состояния" (health_check).

Синтаксис:

CheckMethod ::= health_check_method '(' method ')'

method ::= ping | dns | get

health_result

Результат. Действительно только для условий сценария "Проверка состояния" (health_check).

Синтаксис:

esult ::= health_result '(' result ')'

result ::= positive | negative

health_request_timeout

Тайм-аут подключения (сек). Действительно только для условий сценария "Проверка состояния" (health_check).

Синтаксис:

RequestTimeout ::= health_request_timeout '(' integer ')'

health_answer_timeout

Тайм-аут ответа (сек). Действительно только для условий сценария "Проверка состояния" (health_check), "Метод проверки HTTP GET" (get).

Синтаксис:

ut ::= health_answer_timeout '(' integer ')'

health_type_request

Тип DNS-запроса. Действительно только для условий сценария "Проверка состояния" (health_check), "метод проверки DNS" (dns).

Синтаксис:

TypeRequest ::= health_type_request '(' type_name ')'

type_name ::= word | string

Свойства правил балансировки нагрузки

scheduler

Метод балансировки. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

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

Реальные серверы. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

RealServer ::= real_server '(' mode, ipv4[:port], weight ')'

mode ::= gate | masq | masq_snat

port ::= integer

weight ::= integer

gate — шлюз;

masq — маскардинг;

masq_snat — маскардинг с подменой IP-адреса источника (SNAT).

Пример:

```
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

Аварийный режим сервера. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

IpvsFallback ::= ipvs_fallback '(' mode, ipv4[:port] ')'

mode ::= gate | masq | masq_snat

port ::= integer

monitor_kind

Режим. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorKind ::= monitor_kind '(' kind ')'

kind ::= ping | connect | negotiate

monitor_service

Сервис. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorService ::= monitor_service '(' service ')'

service ::= http | dns

monitor_request

Запрос. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorRequest ::= monitor_request '(' string ')'

monitor_response

Ожидаемый ответ. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorResponse ::= monitor_response '(' string ')'

monitor_interval

Интервал проверки. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorInterval ::= monitor_interval '(' integer ')'

monitor_timeout

Время ожидания. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorTimeout ::= monitor_timeout '(' integer ')'

monitor_failurecount

Число неудачных попыток. Действительно только для правил балансировки нагрузки TCP/UDP.

Синтаксис:

MonitorFailurecount ::= monitor_failurecount '(' integer ')'

Действия

Version: 8.x
Product: DCFW

Действие (action) — это то, что будет выполнено, если условия в правиле истинны. В качестве параметров могут использоваться константные значения, или динамические значения там, где это предусмотрено.

Синтаксис:

```
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

Действия, которые разрешаются без выхода из обработки правил на текущем слое. *warning* ставит отметку о том, что необходимо вставить код с предупреждением в тело ответа.

Пример:

```
[L1]  
category = lib.category(Productivity) warning  
DENY user = user1
```

log_message

Записать сообщения в журнал.

Пример:

```
DENY category = lib.category(Productivity) log_message("Deny porno")
```

append

Добавить заголовок к HTTP-запросу/ответу. Список поддерживаемых заголовков приведен в [Приложении](#).

Первый параметр может быть опущен, если заголовок относится к одной группе *request* или *response*.

Синтаксис:

```
append([request | response,] <headername>, value)
```

headername — см. в [Приложении](#).

value ::= *string* | *numeric* | *condition_name*.

set

Переписать значение конкретному HTTP-заголовку. Список поддерживаемых заголовков см. в [Приложении](#).

Первый параметр может быть опущен, если заголовок относится к одной группе *request* или *response*.

Синтаксис:

```
set([request | response,] <headername>, value)
```

headername — см. в [Приложении](#).

value ::= *string* | *numeric* | *condition_name*.

delete

Удалить HTTP-заголовок. Список поддерживаемых заголовков см. в [Приложении](#).

Первый параметр может быть опущен, если заголовок относится к одной группе *request* или *response*.

Синтаксис:

```
delete([request | response,] <headername>)
```

headername — см. в [Приложении](#).

replace

Модифицировать значение HTTP-заголовка. Список поддерживаемых заголовков см. в [Приложении](#).

Первый параметр может быть опущен, если заголовок относится к одной группе *request* или *response*.

Синтаксис:

```
replace([request | response,] <headername>, regex, value)
```

regex ::= *string* % регулярное выражение

value ::= string | condition_name

headername — см. в [Приложении](#).

Пример 1:

Добавить заголовок Referer:

```
PASS append(Referer, "http://example.com") enabled(true)
```

Удалить заголовок:

```
PASS delete(Referer)
```

Переписать заголовок:

```
PASS set(request, Cache-Control, no-cache)
```

Модифицировать заголовок Location:

```
PASS response.header.Location.count = 1.. replace(response, Location, "http://example.com", url.host) enabled(true)
```

Пример 2:

```
define action delete_referer
  log_message("Referer header deleted")
  delete(request, Referer)
end
```

encrypt

Шифровать часть пути в HTTP-заголовке. Список поддерживаемых заголовков см. в [Приложении](#).

Первый параметр может быть опущен, если заголовок относится к одной группе *request* или *response*.

Ключ шифрования и флаг "Использовать IP как часть ключа шифрования" — необязательные параметры.

Синтаксис:

```
encrypt([request | response,] <headername>, <url>[, <user_key>[, <add_ip>]])
```

url ::= string % часть url для фильтрации

user_key ::= string % пользовательский ключ шифрования (необязательный параметр)

add_ip ::= boolean % добавлять ли IP к ключу шифрования (логическое значение, необязательный параметр)

boolean ::= yes | no | true | false

headername — см. в [Приложении](#).

encrypt_body_url

Шифровать часть пути в ссылках тела ответа.

Ключ шифрования и флаг "Использовать IP как часть ключа шифрования" — необязательные параметры.

Синтаксис:

```
encrypt_body_url(<url>[, <user_key>[, <add_ip>]])
```

url ::= string % часть url для фильтрации

user_key ::= string % пользовательский ключ шифрования (необязательный параметр)

add_ip ::= boolean % добавлять ли IP к ключу шифрования (логическое значение, необязательный параметр)

boolean ::= yes | no | true | false

decrypt_path

Дешифровать часть пути запроса.

Первый параметр может быть опущен, если заголовок относится к одной группе *request* или *response*.

Ключ шифрования и флаг "Использовать IP как часть ключа шифрования" — необязательные параметры.

Синтаксис:

```
decrypt_path(<path>[, <user_key>[, <add_ip>]])
```

path ::= string % часть пути для фильтрации

user_key ::= string % пользовательский ключ шифрования (необязательный параметр)

add_ip ::= boolean % добавлять ли IP к ключу шифрования (логическое значение, необязательный параметр)

boolean ::= yes | no | true | false

Пример:

Шифровать все относительные пути в заголовке *Location* и теле ответа, и дешифровать путь запроса:

```
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

Вставить скрипт в тело ответа.

Синтаксис:

body_inject(inject_text)

inject_text ::= string

set_cookie_token

Добавить в ответ заголовок 'Set-Cookie' со сгенерированным токеном.

Синтаксис:

set_cookie_token(cookie_name, parameter, expires_date)

cookie_name ::= string

parameter ::= string

expires_date ::= [DD_]HH:MM % время которое будет прибавлено к текущему времени

Пример:

Реализация CSRF защиты:

```
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(ucsf_token, "path=/", 01_00:00) body_inject("<script language='JavaScript'>
    var tokenName = 'UCSRF_TOKEN';
```

```
    document.addEventListener('DOMContentLoaded', function()
    {
        var t_res = document.cookie.match(/ucsf_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

Шифровать значения Cookie в заголовке *Set-Cookie* с заданным именем.

Синтаксис:

encode_cookie(cookie_name[, condition_name][, user_kry_string][, f_encrypt])

cookie_name ::= string

condition_name % условие используемое для кодирования (по умолчанию src.ip)

user_kry_string ::= string % пользовательский ключ шифрования (по умолчанию "")

f_encrypt := true % необходимо шифрование (по умолчанию false)

decode_cookie

Дешифровать токен в заголовке Cookie с заданным именем.

Синтаксис:

decode_cookie(cookie_name[, condition_key][, user_kry_string][, f_decrypt])

cookie_name ::= string

condition_name % условие используемое для декодирования (по умолчанию src.ip)

user_kry_string ::= string % пользовательский ключ шифрования (по умолчанию "")

f_encrypt := true % необходимо шифрование (по умолчанию false)

Пример:

Шифрование и дешифрование Cookie с именем *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

Модифицировать тело ответа. Выполняется не более двух (первых) модификаций для каждого ответа.

Синтаксис:

body_replace(<regex>, <value>)

regex ::= string % регулярное выражение

value ::= string

Пример:

```
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

Аутентифицировать пользователя. В случае если IP не указан, запрос маркируется именем пользователя.

Синтаксис:

lookup_and_auth(<user_login>[, <ip_address>[, <session_timeout>]])

user_login ::= string | condition_name % Логин аутентификации

ip_address ::= string | condition_name % IP адрес

session_timeout ::= integer % тайм-аут сессии, по умолчанию 0.

Пример:

```
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

При блокировке перенаправить пользователя на адрес, который указан в редиректе.

Синтаксис:

Redirect ::= redirect(RespCode[, RedirectText], Url)

RespCode ::= 301 | 302 | 305 | 307

RedirectText ::= string

Url ::= string

Пример:

```
DENY src.zone = Trusted redirect(302, "Custom test (Moved)", "https://block.captive/block")
DENY src.zone = Untrusted redirect(302, "https://block.captive/block")
```

inc и dec

Используются для изменения значения переменных, объявленных как *def var*.

Синтаксис:

inc(var.<var_name>, integer)

dec(var.<var_name>, integer)

Пример:

На каждый *http.response.code = 500* увеличивается значение *rps* на 1. Если превысили 10 таких запросов за 5 минут, блокируем дальнейшие ответы. Через 5 минут переменная *rps* будет сброшена в 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

Сбросить значения переменных, объявленных как *init* в *def var*, в начальное значение.

Синтаксис:

reset(var.<var_name>)

sma

Используются для подсчета среднего значения в окне времени, которое определяется в переменной как *window* в *def var*.

Синтаксис:

sma(var.<var_name>, integer)

Пример:

Блокируются запросы, когда среднее время запроса за 30-секундный интервал превысит 2 секунды:

```
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 — подмена сетевых IP-адресов.

Пример:

PASS src.zone = Trusted dst.zone = Untrusted nat

dnat

DNAT — подмена IP-адреса назначения.

Пример:

PASS dnat target_ip("192.168.1.20")

port_mapping

Порт-форвардинг — перенаправление трафика на указанный IP-адрес с изменением номера порта публикуемого сервиса.

Пример:

PASS port_mapping target_ip("192.168.1.20") port_map(tcp, 2000, 2001)

netmap

Network mapping — позволяет произвести замену IP-адресов источника или назначения одной сети на другую.

Пример:

PASS netmap target_ip("192.168.32.0/24") direction(output)

route

Policy-based routing — позволяет маршрутизировать IP-пакеты на основе расширенной информации, например, сервисов, MAC-адресов или серверов (IP-адресов).

SNAT IP — Адрес, на который будет заменен адрес источника для трафика NAT.

Пример:

PASS service = HTTP route gateway(Шлюз1)

ignore

Игнорировать ответ от ICAP-сервера. В этом случае, вне зависимости от ответа ICAP-сервера, данные к пользователю уходят без модификации.

Пример:

OK profile("Example ICAP server") enabled(true) name("ICAP rule") ignore

forward

Переслать. В случае успешной расшифровки трафика SSL/TLS копия трафика будет переслана в соответствии с правилом и профилем инспектирования SSL.

Пример:

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

Правила контентной фильтрации

Префиксы

Имя	Описание
PASS	Разрешение посещения.
DENY	Блокировка веб-страницы.
WARNING	Предупреждение пользователя о том, что страница нежелательна для посещения.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#).
[dst.zone](#), [dst.geoip](#), [dst.ip](#).
[category](#), [scenario](#), [time](#), [url](#), [user](#).
[request.header.Referer](#) — Список URL, в котором указаны рефереры для текущей страницы.
[request.header.User-Agent](#) — Useragent пользовательских браузеров.
[response.header.Content-Type](#) — Списки типов контента.
[http.method](#) — Метод, используемый в HTTP-запросах.

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [virus](#) [heuristic](#).

Пример

```
[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")
```

Правила межсетевого экрана

Префиксы

Имя	Описание
PASS	Разрешение трафика.
DENY	Блокировка трафика.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#).
[dst.zone](#), [dst.geoip](#), [dst.ip](#).
[service](#), [scenario](#), [time](#), [url](#), [user](#), [hip_profile](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [reject_with](#), [ips_profile](#), [I7_profile](#).

Пример

```
[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") \
  I7_profile("Pass all applications") \
  rule_log(yes, "3/h", 5) \
  enabled(true) \
  name("Allow trusted to untrusted")
```

Правила NAT и маршрутизации

Префиксы

Имя	Описание
OK	Всегда ОК.

Тип правила определяется действием ([Action](#)):

- [nat](#) — правило NAT;
- [dnat](#) — правило DNAT;
- [port_mapping](#) — правило порт-форвардинга;
- [netmap](#) — правило Network mapping;
- [route](#) — правило policy-based.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).

[dst.zone](#), [dst.geoip](#), [dst.ip](#).

[service](#), [scenario](#), [time](#), [url](#), [user](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [direction](#), [target_ip](#), [target_snat](#), [snat_target_ip](#),

[port_map](#), [gateway](#).

Пример

```
[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("DNAT 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

```

Правила Captive-портала

Префиксы

Имя	Описание
PASS	Не использовать аутентификацию.
OK	Использовать Captive-профиль.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#).

[dst.zone](#), [dst.geoip](#), [dst.ip](#).

[time](#), [url](#), [category](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [profile](#).

Пример

```

[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")

```

Правила инспектирования SSL

Префиксы

Имя	Описание
PASS	Не расшифровывать передаваемые данные.
OK	Расшифровать передаваемые данные.

Пересылка определяется действием ([Action](#)):

- [forward](#) — в случае успешной расшифровки трафика SSL/TLS копия трафика будет переслана в соответствии с правилом и профилем инспектирования SSL ([ssl_forward_profile](#)).

Условия

[src.zone](#), [src.geoip](#), [src.ip](#),

[dst.geoip](#), [dst.ip](#).

[time](#), [service](#), [user](#), [category](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [ssl_profile](#), [ssl_forward_profile](#).
[block_invalid_cert](#), [check_revoc_cert](#), [block_expired_cert](#), [block_self_signed_cert](#).

Пример

```
[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
```

Правила инспектирования SSH

Префиксы

Имя	Описание
PASS	Не расшифровывать передаваемые данные.
OK	Расшифровать передаваемые данные.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#),
[dst.geoip](#), [dst.ip](#),
[time](#), [service](#), [user](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#).
[block_ssh_shell](#), [block_ssh_exec](#), [block_sftp](#), [ssh_command](#).

Пример

```
[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-правила

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[url.domain](#).

Свойства

[name](#), [desc](#), [enabled](#), [dns_server](#).

Пример

```
[dns "DNS Rules"]
% ----- 1 -----
OK \
  url.domain = "*.example.com" \
  dns_server(1.2.3.4) \
  enabled(true) \
  name("Dns rule")
```

DoS-правила

Префиксы

Имя	Описание
PASS	Разрешение трафика.
DENY	Безусловная блокировка трафика.
WARNING	Применение профиля защиты от DoS-атак.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#),
[dst.zone](#), [dst.geoip](#), [dst.ip](#).
[time](#), [service](#), [user](#), [scenario](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [profile](#).

Пример

```
[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-правила

Префиксы

Имя	Описание
PASS	Пропуск. Не посылать данные на ICAP-сервер.
OK	Пересылка данных на ICAP-сервер.

Действие ([Action](#)):

- [ignore](#) — игнорировать ответ от ICAP-сервера.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#).
[dst.zone](#), [dst.geoip](#), [dst.ip](#).
[url](#), [category](#), [user](#), [service](#), [http.method](#), [response.header.Content-Type](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [profile](#).

Пример

```
[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")
```

Правила защиты почтового трафика

Префиксы

Имя	Описание
PASS	Пропуск трафика без изменений.
DENY ("with error")	Блокировка письма, при этом сообщается об ошибке доставки письма на сервер.
DENY	Блокировка письма без уведомления о блокировке.
WARNING	Маркировка почтовых сообщений.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#).
[dst.zone](#), [dst.geoip](#), [dst.ip](#).
[user](#), [service](#), [envelope_from](#), [envelope_to](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [mark_hdr](#), [mark](#), [antispam_usergate](#), [dnsbl](#).

Пример

```
[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")
```

Правила reverse-прокси

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).

[dst.zone](#), [dst.geoip](#), [dst.ip](#).

[user](#), [request.header.User-Agent](#), [url.port](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [profile](#), [certificate](#), [cert_auth_enabled](#), [is_https](#), [ssl_profile](#),

[waf_profile](#), [rewrite_path](#).

Пример

```
[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")
```

Правила веб-безопасности

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).

[dst.zone](#), [dst.geoip](#), [dst.ip](#).

[time](#), [user](#),

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#).

[enable_adblock](#), [safe_search](#), [search_history_logging](#), [social_sites_block](#), [enable_injector](#).

[custom_injector](#), [url_list_exclusions](#).

Пример

```
[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")
```

Правила пропускной способности

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).

[dst.zone](#), [dst.geoip](#), [dst.ip](#).

[time](#), [user](#), [application](#), [service](#), [scenario](#).

Свойства

[name](#), [desc](#), [enabled](#), [rule_log](#), [bandwidth_pool](#).

Пример

```
[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")
```

Правила инспектирования туннелей

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).

[dst.zone](#), [dst.geoip](#), [dst.ip](#).
[service](#).

Свойства

[name](#), [desc](#), [enabled](#).

Пример

```
[tunnel "Tunnel inspection Rules"]
% ----- 1 -----
PASS \
  src.zone = Trusted \
  dst.zone = Untrusted \
  service = gre \
  enabled(true) \
  name("Example Tonnel Inspection Bypass rule")
% ----- 2 -----
OK \
  dst.geoip = YE \
  service = gtpu \
  enabled(true) \
  name("Example Tonnel Inspection Rule")
```

Правила веб-портала

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[url](#), [user](#), [url.domain](#).

Свойства

[name](#), [desc](#), [enabled](#).

[icon](#), [ssl_profile](#), [certificate](#), [additional_url](#), [rdp_check_session_alive](#), [transparent_auth](#).

Пример

```
[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")
```

Правила VPN-сервера

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).
[dst.geoip](#), [dst.ip](#).
[user](#).

Свойства

[name](#), [desc](#), [enabled](#).

[profile](#), [auth_profile](#), [vpn_network](#), [interface](#).

Пример

```
[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")
```

Правила VPN-клиента

Префиксы

Имя	Описание
OK	Всегда ОК.

Свойства

[name](#), [desc](#), [enabled](#).
[profile](#), [interface](#), [server_address](#).
[last_error](#), [status](#), [connection_time](#) — информационные поля, недоступные для редактирования.

Пример

```
[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")
```

Правила сценариев

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[scenario_cond](#).

Свойства

[name](#), [desc](#), [enabled](#).
[operation_mode](#), [trigger](#), [duration](#).

Условия сценария

Категория URL (url_category)

Условия

[category](#)

Свойства

[count_interval](#), [max_event_count](#), [scond_type](#)

Обнаружение вируса (virus_detection)

Свойства

[scond_type](#)

Приложение (app)

Условия

[application](#)

Свойства

[count_interval](#), [max_event_count](#), [scond_type](#)

СОВ (ips)

Свойства

[ips_tl](#), [scond_type](#)

Тип котента (mime_type)

Условия

[response.header.Content-Type](#)

Свойства

[count_interval](#), [max_event_count](#), [scond_type](#)

Размер пакета (net_packet_size)

Свойства

[packet_size](#), [scond_type](#)

Сессий с одного IP (sessions_per_ip)

Свойства

[sessions_limit](#), [scond_type](#)

Объем трафика (traffic)

Свойства

[traffic_limit](#), [period](#), [scond_type](#)

Проверка состояния (health_check)

Условия

[url.address](#), [url.domain](#)

Свойства

[health_check_method](#), [health_result](#), [health_request_timeout](#), [health_type_request](#),

[health_answer_timeout](#), [count_interval](#), [max_event_count](#), [scond_type](#)

Пример

```
[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")
```

Правила балансировки нагрузки TCP/UDP

Префиксы

Имя	Описание
OK	Всегда ОК.

Условия

[src.zone](#), [src.geoip](#), [src.ip](#), [src.mac](#).

[service](#), [url.address](#), [url.port](#).

Свойства

[name](#), [desc](#), [enabled](#).

[scheduler](#), [real_server](#), [ipvs_fallback](#)

[monitor_kind](#), [monitor_service](#), [monitor_request](#), [monitor_response](#), [monitor_interval](#),

[monitor_timeout](#), [monitor_failurecount](#)

Пример

```
[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")
```

Правила балансировки ICAP

Префиксы

Имя	Описание
OK	Всегда ОК.

Свойства

[name](#), [desc](#), [enabled](#), [profile](#).

Пример

```
[icap_balancing "ICAP load balancing Rules"]
% ----- 1 -----
OK \
  profile("Example ICAP server") \
  enabled(true) \
  name("ICAP load balancing")
```

Правила балансировки reverse-прокси

Префиксы

Имя	Описание
OK	Всегда ОК.

Свойства

[name](#), [desc](#), [enabled](#), [profile](#).

Пример

```
[reverseproxy_balancing "Reverse proxy load balancing Rules"]
```

```
% ----- 1 -----
```

```
OK \
```

```
  profile("Example reverse proxy server") \
```

```
  enabled(true) \
```

```
  name("Reverse-proxy load balancing")
```

Список поддерживаемых HTTP-заголовков

Version: 8.x
Product: DCFW

HTTP Header	Request/Response	SET/REPLACE/ENCRYPT	APPEND	DELETE
Accept	Request	+	+	+
Accept-Charset	Request	+	+	+
Accept-Encoding	Request	+	+	+
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	+		
Expires	Request/Response		+	+
From	Request	+	+	
Host	Request			
If-Match	Request	+		
If-Modified-Since	Request			
If-None-Match	Request	+		
If-Range	Request			

HTTP Header	Request/Response	SET/REPLACE/ENCRYPT	APPEND	DELETE
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	+	+	+
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			

Список категорий

Version: 8.x

Product: DCFW

ID	Название	Description	Описание
0	Unknown	Not categorized	Сайты без принадлежности к какой-либо категории.
1	Advertisements and Pop-Ups	Sites that provide advertising graphics or other ad content files that appear on Web pages.	Сайты, содержащие графическую рекламу (баннеры, всплывающие окна и прочее).
2	Alcohol and Tobacco	Sites that promote or sell alcohol- or tobacco-related products or services.	Сайты, посредством которых реализуется продвижение или продажа алкогольной и табачной продукции
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.	Сайты, предоставляющие возможность анонимного посещения других сайтов и ресурсов интернета. Анонимайзеры также могут применяться для обхода средств веб-фильтрации и т.д.
4	Arts	Sites with artistic content or relating to artistic institutions such as theaters, museums, galleries, dance companies, photography, and digital graphic resources.	Сайты, посвященные искусству, в том числе различным театрам, музеям, галереям, фотографии, танцевальным коллективам и т.д.
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.	К данной категории относятся корпоративные веб-сайты, а также интернет-ресурсы, содержащие информацию об услугах и продуктах, необходимых предприятиям любого размера для успешной деятельности.
6	Transportation	Sites that include information about 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.	Сайты, посвященные автотранспорту и его покупке. Категория включает в себя интернет-ресурсы производителей, дилеров, сайты с обзорами, сообщества, клубы автолюбителей и т.д.
7	Chat	Sites that enable web-based exchange of real-time messages through chat services or chat rooms	Сайты, позволяющие осуществлять общение в режиме реального времени.
8	Kids sites	Websites that are family and children oriented.	Сайты, ориентированные на детскую аудиторию
9	Forums and Newsgroups	Sites for sharing information in the form of newsgroups, forums, bulletin boards. Does not include personal blogs.	Ленты новостей, форумы, доски объявлений. Категория не включает в себя персональные блоги.
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.	Сайты, контроль над которыми был захвачен злоумышленниками для распространения вредоносных программ без ведома владельцев. В категорию также входят ресурсы из группы высокого риска заражения.
11	Computers and Technology	Sites that contain information such as product reviews, discussions, and news about computers, software, hardware, peripheral and computers services.	Сайты, содержащие обзоры продуктов, новости и информация о компьютерах, программном обеспечении, оборудовании, периферийных устройствах, технологиях и услугах.
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.	Сайты, предлагающие советы и рекомендации по совершению противоправных или преступных действий, а также содержащие информацию о том, как скрыть следы преступления. На подобных ресурсах можно узнать о том, как совершить убийство, создать бомбу, взломать замки и т.д. В данную категорию также включены сайты о незаконном использовании электронных устройств, хакерстве, мошенничестве и незаконном распространении программного обеспечения.
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.	Сайты, способствующие налаживанию межличностных отношений, таких как знакомства и брак. Категория включает в себя сайты для сватовства, знакомств, ресурсы, предлагающие эскорт-услуги.
14	Download Sites	Sites that contain downloadable software, whether shareware, freeware, or for a charge. Includes some peer-to-peer sites.	Сайты, которые содержат доступное для скачивания программное обеспечение. Данная категория также включает в себя некоторые Peer-to-Peer сайты.

ID	Название	Description	Описание
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	Сайты, поддерживаемые учебными заведениями и школами всех типов, включая дистанционное обучение. Включает в себя общеобразовательные и справочные материалы, такие как словари, энциклопедии, интерактивные курсы, учебные пособия и методические материалы.
16	Entertainment	Sites containing programming guides to television, movies, music and video (including video on demand), celebrity sites, and entertainment news.	Сайты, предлагающие пользователю телеканалы, кино, музыку и видео (в том числе видео по запросу). Категория также включает в себя сайты знаменитостей и развлекательные новости.
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 companies, and so on.	Сайты, связанные с банками, финансами, платежами или инвестициями. В категорию входят интернет-ресурсы банков, брокерских компаний, порталы для онлайн-торговли на бирже. На данных ресурсах также можно узнать котировки акций, информацию об управлении фондами, страховых компаниях, кредитных союзах и т.д.
18	Gambling	Sites that offer or are related to online gambling, lottery, casinos and betting agencies involving chance.	Сайты, связанные с азартными играми, лотереями, казино и букмекерскими агентствами.
19	Games	Sites relating to computer or other games, information about game producers, or how to obtain cheat codes. Game-related publication sites.	Сайты, связанные с компьютерными или другими видами игр. Ресурсы содержат информацию о разработчиках игр, кодах и читках.
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.	Сайты, находящиеся в ведении правительственных или военных организаций, ведомств или учреждений, в том числе отделений полиции, пожарной охраны, таможенных бюро, аварийных служб, гражданской обороны, служб борьбы с терроризмом и больниц.
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, age, disability, sexual orientation or nationality.	Интернет-ресурсы, посвященные вопросу поощрения ущемления прав людей в зависимости от расы, религии, пола, возраста, инвалидности, сексуальной ориентации или национальности.
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	Сайты, содержащие информацию, относящуюся к поддержанию здоровья, медицинским услугам, фитнесу и благополучию. В данную категорию также включены интернет-ресурсы, связанные с медицинским оборудованием, больницами, аптеками, сестринским делом, медициной, рецептами лекарств, и т.д.
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.	Сайты с информацией о приобретении, производстве и использовании незаконных или рекреационных наркотиков, а также злоупотреблении отпускаемыми по рецепту лекарствами.
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.	Сайты, содержащие списки вакансий, оказывающие помощь в поиске работы. В категорию входят интернет-ресурсы агентств по трудоустройству и рекрутинговых компаний.
25	Military	Information on military branches, armed services, and military history.	Информация о военных организациях, военной истории.
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.	Сайты, содержащие потоковый контент, например, интернет-радио, интернет-телевидение или MP3-live. Категория включает в себя фан- или официальные сайты музыкантов, групп или звукозаписывающих компаний.
27	News	Sites covering news and current events such as newspapers, newswire services, personalized news services, broadcasting sites, and magazines.	Новостные сайты: онлайн-журналы, новостные ленты и т.д.
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.	Сайты, посвященные клубам, сообществам, союзам и некоммерческим организациям. Значительное число данных структур созданы в образовательных или благотворительных целях.
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 apparel, or swimwear.	Ресурсы, содержащие изображение частичной или полной наготы. Категория включает в себя сайты, рекламирующие или продающие нижнее белье, интимную одежду или купальные костюмы.

ID	Название	Description	Описание
30	Personal Sites	Sites about or hosted by personal individuals, including those hosted on commercial sites such as Blogger, AOL, etc.	Персональные сайты, созданные отдельными людьми или группами лиц, блоги.
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	Сайты, которые используются для обмана или в мошеннических целях (например, фишинг), такие как кражи финансовой или другой информации посредством учетной записи пользователя. Эти сайты чаще всего являются искусно выполненной подделкой популярных законных ресурсов, что вводит в заблуждение пользователей.
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.	Сайты, посвященные политическим партиям или занимающиеся политической пропагандой, а также предоставляющие информацию о группах интересов, выборах, законодательстве или вопросе лоббирования. Категория включает в себя сайты, которые предлагают сведения о правах и их защите.
33	Pornography/Sexually Explicit	Sites that contain explicit sexual content. Includes adult products such as sex toys, CD-ROMs, and videos, adult services such as videoconferencing, escort services, and strip clubs, erotic stories and textual descriptions of sexual acts.	Сайты, которые содержат контент сексуального содержания. Категория включает в себя продукцию для взрослых, такую как секс-игрушки, компакт-диски и видео. Сайты из данной группы позволяют заказывать эскорт-услуги, получить сведения о стриптиз-клубах, почитать эротические рассказы и текстовые описания половых актов.
34	Real Estate	Sites relating to commercial or residential real estate services, including renting, purchasing, selling or financing homes, offices, etc.	Информация об аренде, покупке или продаже недвижимости или земельных участков. Рекомендации по покупке или продаже жилых помещений. Агентства недвижимости, услуги по аренде, переезду, улучшению жилищных условий.
35	Religion	Sites that deal with faith, human spirituality or religious beliefs, including sites of churches, synagogues, mosques and other houses of worship.	Сайты, посвященные вопросу веры, человеческой духовности и религиозным убеждениям, в том числе сайты церквей, синагог, мечетей и других молитвенных домов.
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.	Сайты, содержащие списки, обзоры заведений общественного питания. Категория включает в себя сайты с рецептами, инструкциями и советами, продуктами питания и советами по выбору вина.
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.	Сайты, позволяющие осуществлять поиск в интернете по новостям, изображениям, каталогам и другим видам онлайн-контента. Категория включает в себя порталы и каталоги сайтов, такие как белые/желтые страницы.
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.	Интернет-магазины, каталоги, аукционы, рекламные объявления. Категория не включает в себя ресурсы, посвященные товарам и услугам из более узких категорий, таких как здравоохранение и медицина.
39	Social Networking	Sites that enable social networking for online communities of various topics, for friendship, dating, or professional reasons.	Сайты социальных сетей, пользователи которых могут общаться, взаимодействовать, обмениваться сообщениями, файлами.
40	Spam Sites	Sites that have been promoted through spam techniques.	Сайты, продвижение которых основано на использовании спам-рассылок.
41	Sports	Sites relating to sports teams, fan clubs, scores and sports news. Relates to all sports, whether professional or recreational.	Сайты команд или спортивных ассоциаций, международные и национальные, спортивные колледжи. Ресурсы, содержащие результаты и расписания игр, спортивные журналы и газеты, виртуальный спорт и спортивные лиги.
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.	Сайты, содержащие нежелательные программы, автоматически загружающиеся и устанавливающиеся на ПК пользователей. Вирусы вносят изменения в систему и позволяют злоумышленникам получить данные с компьютера без ведома и согласия его владельца.
43	Stock trading	Promotion and facilitation of securities trading and management of investment assets. Also includes information on financial investment strategies, quotes, and news.	Сайты по торговле ценными бумагами и управления инвестиционными активами. Также включает информацию о финансово-инвестиционной стратегии, котировкам и новостям.
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	Сайты, на которых возможен онлайн-перевод веб-страниц на другие языки. Данные ресурсы могут использоваться для обхода заданных ограничений, так как просмотр сайтов осуществляется с URL сайта онлайн-переводчика.

ID	Название	Description	Описание
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.	Авиакомпании, агентства по бронированию авиабилетов, планирование поездок, предварительные заказы, аренда транспортных средств, описания мест назначения, реклама гостиниц и казино.
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.	Сайты, содержащие изображения, текстовые описания и пропаганду физического насилия против людей, животных или организаций/сообществ. Категория включает в себя ресурсы с иллюстрациями ужасов природы, а также сайты, содержащие ненормативную лексику.
47	Weapons	Sites that depict, sell, review or describe guns and weapons, including for sport.	Сайты, содержащие изображения, обзоры и описания оружия. В категории также входят интернет-магазины орудий и оружия, в том числе для занятия спортом.
48	Web-based Email	Sites that enables users to send and receive email through a web-accessible email account.	Порталы, позволяющие завести собственные почтовые аккаунты для написания, чтения, отправки и получения электронных писем.
49	General	Sites that do not clearly fall into other categories, for example, blank web pages	Сайты, которые не попадают в другие категории, например, пустые веб-страницы.
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.	Сайты, связанные с рекреационной деятельностью и увлечениями, включая ресурсы, посвященные зоопаркам, центрам общественного отдыха, бассейнам, паркам развлечений и хобби, таким как садоводство, литература, искусства и ремесла, домашний декор и т.д.
51	Online training and tools	Distance education, online courses, vocational training, software training, skills training.	Дистанционное образование, онлайн-курсы, профессиональное обучение, программы обучения, повышения квалификации.
52	Legal	Legal websites, law firms, discussions and analysis of legal issues.	Сайты юридических фирм, обсуждение и анализ юридических вопросов.
53	Local Information	City guides and tourist information, including restaurants, area information and local points of interest.	Путеводители и информация для туристов, в том числе о ресторанах и местные достопримечательности.
54	Reference and Research	Personal, professional, or educational reference material, including online dictionaries, maps, library catalogues and scientific information.	Профессиональные или образовательные справочные материалы, в том числе онлайн-словари, карты, переписи, альманахи, каталоги библиотек, генеалогии, и научной информации.
55	Technical or business forums and news groups	Websites with discussions of user-generated content related to business or technical development.	Сайты с обсуждениями, относящимся к деловой или технической тематикам.
56	Technical information and documentation	Websites that provide information on technical information and documentation.	Сайты, которые предоставляют информацию к технической документации и информации.
57	Personal Storage	Websites that allow the uploading of files for remote data storage.	Сайты, которые позволяют загружать файлы и резервные копии для удаленного хранения данных.
58	CDNs	Content distribution networks	Сети предоставления контента
59	Profanity	Websites which contain excessive use of profanity or obscenities.	Сайты, содержащие нецензурную лексику
60	Professional social networks	The subset of social networking websites which includes content intended exclusively for businesses or professionals.	Подмножество веб-сайтов социальных сетей, которое включает в себя контент, предназначенный исключительно для бизнеса и профессионалов.
61	Botnets	Sites that manage networks of bots through command-and-control centers.	Сайты, которые управляют сетями ботов через командно-контрольные центры.
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.	Сайты, связанные с нетрадиционной религиозной практикой, обычно называемой культами. Подобные движения считаются ложными, неортодоксальными, экстремистскими. Члены культов зачастую живут под руководством харизматического лидера.

ID	Название	Description	Описание
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.	Сайты о моде, ювелирных изделиях, гламуре, красоте, модельном бизнесе, косметике или связанных со всем перечисленным товарах или услугах. Включает отзывы, сравнения, а также общую информацию, полезную потребителю.
64	Greeting cards	Sites that allow people to send and receive greeting cards and postcards.	Сайты, которые позволяют людям получать и отправлять поздравительные открытки.
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	Сайты, которые способствуют или содержат рекомендации о том, как получить несанкционированный доступ к компьютерным системам с целью кражи информации, совершения мошеннических действий, советы по созданию вирусов или совершению других видов незаконной деятельности, связанной с кражей цифровой информации.
66	Cryptocurrency Mining	Sites that use cryptocurrency mining technology without user permission.	Сайты, которые используют технологии майнинга криптовалют без разрешения пользователя.
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.	Сайты, которые незаконно распространяют программное обеспечение или защищенные авторским правом материалы, такие как фильмы, музыка, незаконные серийные номера, генераторы ключей.
68	Image Sharing	Sites that host digital photographs and images, online photo albums and digital photo exchanges.	Порталы для размещения цифровых фотографий и изображений, фотоальбомы и фотообменники.
69	Information Security	Sites that provide legitimate information about data protection, including newly discovered vulnerabilities and how to block them	Сайты, которые предоставляют информацию о законной защите данных, в том числе о новых угрозах и способах борьбы с ними.
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	Сайты, которые позволяют создавать и использовать аккаунт в службах обмена мгновенными сообщениями, таких как ICQ, AOL Instant Messenger, IRC, MSN, Jabber, Yahoo Messenger, и т.п.
71	Network Errors	Sites that do not resolve to any IP address.	Сайты, которым не присвоен IP-адрес.
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.	К паркованным доменам относятся URL-адреса с ограниченным контентом, конечные точки перехода с рекламных объявлений. Подобные ресурсы могут приносить прибыль, но, как правило, не несут в себе полезной для пользователя информации. Категория также включает в себя сайты, находящиеся на реконструкции, папки и базовые страницы веб-серверов.
73	Peer-to-Peer	Sites that enable direct exchange of files between users without dependence on a central server.	Сайты, которые позволяют осуществлять прямой обмен файлами между пользователями вне зависимости от центрального сервера.
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.	Включает наборы IP-адресов, определенные в 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.	Сайты с ответами на различные школьные тесты, написанными эссе, рефератами, исследовательскими или курсовыми работами.
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.	Сайты, связанные с сексуальным образованием, включая такие темы, как уважение к партнеру, аборт, бисексуальный образ жизни, противозачаточные средства, заболевания, передаваемые половым путем и беременность.
77	Tasteless	Sites with offensive or tasteless content such as bathroom humor, or gruesome or even frightening content such as shocking depictions of blood or wounds, or cruel animal treatment.	Сайты с оскорбительным контентом, такого рода как юмор "ниже пояса", ужасный или даже пугающий контент, шокирующие изображения крови, ран или жестокого обращения с животными.
78	Child Abuse Images	Sites that portray or discuss children in sexual or other abusive acts	Сайты, содержащие фотографии, иллюстрирующие случаи жестокого обращения с детьми.
79	Gay, Lesbian or Bisexual	Web pages that cater to or discuss the gay, lesbian, bisexual or transgender lifestyle.	Сайты, содержащие материалы о геех, лесбиянках, трансгендерах.

ID	Название	Description	Описание
80	Literature and Books	Web pages for published writings including fiction and non-fiction novels, poems and biographies.	Сайты, содержащие опубликованные книги, новеллы, рассказы, включая фантастику, биографии.
81	Nutrition and Diet	Web pages on losing weight and eating healthy, diet plans, weight loss programs and food allergies.	Сайты, посвященные правильному питанию, снижению веса, диетам.
82	Pets and Animals	Web pages with information or products and services for pets and other animals including birds, fish, and insects.	Сайты, посвященные продуктам и сервисам для домашних и других животных.
84	Updates	All sites with software updates, OS updates, security updates.	Сайты обновлений программного обеспечения, обновлений операционных систем, обновлений
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.	Сайты с низким репутационным риском. Как правило это сайты, не имеющие известных уязвимостей, но относящиеся к категориям, где риск нарушения безопасности возможен.
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.	Сайты со средним репутационным риском. Как правило это сайты, которые могут содержать уязвимости и относящиеся к категориям, где риск нарушения безопасности возможен.
99	Reputation: High risk	Sites with high reputational risk. Web sites that are confirmed to be a security risk.	Сайты с высоким репутационным риском. Сайты с известными уязвимостями.

