Table of Contents

UserGate Policy Language (UPL) **1**
UserGate Policy Language (Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ) 1
Ð Ð±Ñ Ð¸Ðµ Ð¿Ð¾Ð»Ð¾Ð¶ÐµÐ½Ð¸Ñ 4
Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ 7
Ð Ñ Ñ Ñ Ð¾ÐµÐ½Ð½Ñ Ðµ Ð±Ð¸Ð»Ð¸Ð¾ÐµÐºÐ¸ 18
Ð Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ñ 19
ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° 21
ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL block_invalid_cert 24
ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° BYOD 32
max_device_number 32
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ 42
Ð¢Ð¸Ð¿Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» 52
ÐiÐ¿Ð¸Ñ Ð¾Ð¼ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð² 70

UserGate Policy Language (UPL)

UserGate Policy Language (Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ)

Version: 7.x
Product: NGFW
UPL (UserGate Policy Language) â Ñ Ð·Ñ Ðº Ð¾Ð¿Ð¸Ñ Ð°Ð½Ð¸Ñ Ð¿Ð¾Ð»Ð¸Ñ Ð¸Ðº UserGate. Ð¢ÐµÑ Ð¼Ð¸Ð½ "Ð¿Ð¾Ð»Ð¸Ñ Ð¸ÐºÐ°" Ñ Ð¿Ð¾Ñ Ñ ÐµÐ±Ð»Ñ ÐµÑ Ñ Ð·Ð´ÐµÑ Ñ Ð² ÐºÐ¾Ð½Ñ ÐµÐºÑ Ñ Ðµ ÐºÐ¾Ð½Ñ Ð¸Ð³ÑƒÑ€Ð°Ñ†Ð¸Ð¸, Ð¿Ñ Ð¸Ð¼ÐµÐ½Ñ ÐµÐ¼Ñ Ñ Ð´Ð»Ñ Ð¿Ð½Ñ Ñ ÐµÑ Ð¿Ð¸Ñ Ð°Ð²Ð¸Ð» Ð¿Ð¾ Ñ ÐµÑ Ð¿Ð¸Ñ Ð°Ð½Ð¸Ñ Ð¼ Ð°Ñ Ñ ÐµÐ±Ð¾Ð²Ð°Ð½Ð¸Ñ Ð¼ Ð°Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ð¼ Ð´Ð¾Ð»Ð¶Ð½Ñ Ð¾Ð±Ñ Ð·Ð°Ñ‚ÐµÐ»Ñ Ð½Ñ Ð¾Ð±Ñ Ð¾Ð¼ Ð¸Ð¼ÐµÐ½Ð¸ Ð¸Ð¼ÐµÐ½Ð¸ Ð°.

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð½Ð°Ñ Ñ Ñ Ð°Ð¸Ð²Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ‚ÐµÐ»Ñ Ð¼ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ð², Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ð¸Ñ Ð²Ð¾Ð¹Ñ Ñ Ð².

Ð Ð»Ñ ÐºÐ°Ð¶Ð´Ð¾Ð³Ð¾ Ð¿Ð°Ñ€Ð°Ð¼ÐµÑ‚Ñ€Ð° Ð¿Ð¾Ð»Ð° Ð½Ð°Ñ Ñ Ð°Ð¸Ð²Ð°Ñ Ñ Ð¸Ð¼Ñ Ð¸ Ð¸Ð¼Ñ Ð¸Ð¼ÐµÐ½Ð¸ Ð¸Ð¼ÐµÐ½Ð¸ Ð¸Ð¼ÐµÐ½Ð¸

- NAT Ð¸Ð°Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ñ (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **network-policy nat-routing**).
- Ð Ñ Ð¾Ð¿Ñ Ñ ÐºÐ½Ð°Ñ Ñ Ð¿Ð¾Ñ Ð¾Ð±Ñ Ð¾Ð±Ñ Ñ Ñ (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **network-policy traffic-shaping**).
- Ð¤Ð¸Ð»Ñ Ñ Ñ Ð°Ñ Ð¸Ñ ÐºÐ¾Ð½Ñ ÐµÐ½Ñ Ð° (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy content-filtering**).
- Ð ÐµÐ±-Ð±ÐµÐ·Ð¾Ð¿Ð°Ñ Ð½Ð¾Ñ Ñ Ñ (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy safe-browsing**).
- Ð Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ðµ Ñ Ñ Ð½Ð½ÐµÐ»ÐµÐ¹ (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy tunnel-inspection**).
- Ð Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ðµ SSL (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy ssl-inspection**).
- Ð Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ðµ SSH (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy ssh-inspection**).
- Ð Ð°Ñ Ð¸Ñ Ð° Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð¹ Ñ Ñ Ð°Ñ Ð¸ÐºÐ° (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy mail-security**).
- ICAP-Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy icap-rules**).
- Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð·Ð°Ñ Ñ DoS (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **security-policy dos-rules**).
- Ð ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð» (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **global-portal web-portal**).
- Ð Ñ Ð°Ð²Ð¸Ð»Ð° reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸ (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **global-portal reverse-proxy-rules**).
- ÐiÐµÑ Ð²ÐµÑ Ð½Ñ Ðµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° VPN (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **vpn server-rules**).
- Ð Ð»Ð¸ÐµÐ½Ñ Ñ ÐºÐ¸Ðµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° VPN (Ñ Ñ Ð¾Ð²ÐµÐ½Ñ : **vpn client-rules**).

ÐiÑ Ñ Ñ ÐºÑ Ñ Ñ Ð° ÐºÐ¾Ð¼Ð¼Ð°Ð½Ð´Ñ Ð´Ð»Ñ Ñ Ð¾Ð·Ð´Ð°Ð½Ð¸Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°:

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

Ð³Ð´Ðµ <level> â Ñ Ñ Ð¾Ð²ÐµÐ½Ñ , Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ð¾Ñ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾ Ñ Ð¾Ð·Ð´Ð°Ñ Ñ Ð¿Ð°Ð²Ð¸Ð»Ð¾.

<position> â Ð¿Ð¾Ð·Ð¸Ñ , Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ð±Ñ Ð´ÐµÑ Ð¿Ð¾Ð¼ÐµÑ ÐµÐ½Ð¾ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾.

<str-upl-syntax> Ñ Ñ Ð¾ÐºÐ°, Ð² ÐºÐ¾Ñ Ð¾Ñ Ð¾Ñ Ð¿Ð¾Ð¸ Ð°Ð½Ð½Ð¾Ð¹ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ð² UPL Ñ Ð¸Ð½Ñ Ð°Ð½Ñ Ð¸Ñ Ðµ.

ÐiÑ Ñ Ñ ÐºÑ Ñ Ñ Ð° ÐºÐ¾Ð¼Ð¼Ð°Ð½Ð´Ñ Ð´Ð»Ñ Ð¾Ð±Ð½Ð¾Ð²Ð»ÐµÐ½Ð¸Ñ Ñ Ñ Ñ Ðµ Ñ Ð²Ñ Ñ Ñ Ðµ Ð³Ð¾ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°:

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

Ð³Ð´Ðµ <level> â Ñ Ñ Ð¾Ð²ÐµÐ½Ñ , Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ð¾Ñ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾ Ð¾Ð±Ð½Ð¾Ð²Ð¸Ñ Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾.

<position> â Ð½Ð¾Ð²Ð°Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°, ÐºÐ¾Ñ Ð¾Ñ Ð¾Ðµ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾ Ð¾Ð±Ð½Ð¾Ð²Ð¸Ñ Ñ .

<str-upl-syntax> Ñ Ñ Ð¾ÐºÐ°, Ð² ÐºÐ¾Ñ Ð¾Ñ Ð¾Ñ Ð¿Ð¾Ð¸ Ð°Ð½Ð½Ð¾Ð¹ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ð² UPL Ñ Ð¸Ð½Ñ Ð°Ð½Ñ Ð¸Ñ Ðµ.

ÐiÑ Ñ Ñ ÐºÑ Ñ Ñ Ð° ÐºÐ¾Ð¼Ð¼Ð°Ð½Ð´Ñ Ð´Ð»Ñ Ñ Ð´Ð°Ð»ÐµÐ½Ð¸Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°:

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

Ð³Ð´Ðµ <level> â Ñ Ñ Ð¾Ð²ÐµÑ , Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ð¾Ð¼
Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ñ Ñ Ð´Ð°Ð»Ñ Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾.

<position> â Ð½Ð¾Ð¼ÐµÑ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°, ÐºÐ¾Ñ Ð¾Ñ Ð¾Ðµ
Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ñ Ñ Ð´Ð°Ð»Ñ Ñ .

<all> â Ñ Ð´Ð°Ð»Ñ Ñ Ð²Ñ Ðµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°.

Ði Ñ Ñ ÐºÑ Ñ Ñ Ð° ÐºÐ¾Ð¼Ð°Ð½Ð´Ñ Ð´Ð»Ñ Ð¾Ñ Ð¾Ð±Ñ Ð°Ð¶Ð¸Ð²Ð°Ñ
Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°:

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

Ð³Ð´Ðµ <level> â Ñ Ð°Ð·Ð´ÐµÐ», Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° ÐºÐ¾Ñ Ð¾Ñ Ð¾Ð³Ð¾
Ð½Ñ Ð¶Ð½Ð¾ Ð¾Ñ Ð¾Ð±Ñ Ð°Ð·Ð¸Ñ Ñ .

<position> â Ð½Ð¾Ð¼ÐµÑ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°, ÐºÐ¾Ñ Ð¾Ñ Ð¾Ðµ
Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ñ Ð¾Ñ Ð¾Ð±Ñ Ð°Ð·Ð¸Ñ Ñ .

<all> â Ð¾Ñ Ð¾Ð±Ñ Ð°Ð·Ð¸Ñ Ñ Ð²Ñ Ðµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°.

Ð Ñ Ð¸Ð¼ÐµÑ Ñ Ð¾Ð·Ð´Ð°Ð½Ð¸Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾
Ñ ÐºÑ Ð°Ð½Ð° Ñ Ð¸Ð¿Ð¾Ð»Ñ Ð·Ñ Ð¹Ñ Ð²Ð°Ð¹ UPL (Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ñ Ð²Ð°Ð²Ð°Ð½Ð¸Ñ
Ð¼Ð½Ð¾Ð³Ð¾Ñ Ñ Ñ Ð¾Ñ Ð½Ñ Ð¹ Ð²Ð²Ð¾Ð´Ð°):

```
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)
```

Ð Ð¾Ñ Ð»Ðµ Ñ Ð¾Ð·Ð´Ð°Ð½Ð¸Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ð¾Ñ Ð¾Ð±Ñ Ñ Ñ Ð²
Ð½Ð°Ñ Ð°Ð»Ðµ Ñ Ð¿Ñ ÐºÐ° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾
Ñ ÐºÑ Ð°Ð½Ð° (Ð½Ð° Ð¿Ð¾Ð¼Ð¾Ñ Ð¸Ñ Ð°Ð²Ð¸Ð»Ð¾ 1). Ð Ð°Ð½Ð½Ð¾Ðµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾
Ð·Ð°Ð¿Ñ ÐµÑ Ð°ÐµÑ HTTPS-Ñ Ñ Ð°Ñ Ð¸Ð¼ Ð¸Ð· Ð¾Ð¾Ð½Ñ Trusted Ð² Ð·Ð¾Ð½Ñ
Untrusted Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÑ Ñ,
Ð¸Ð´ÐµÐ½Ñ Ð¸Ñ Ð¸Ñ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ Ð¼ Ñ Ð¸Ñ Ñ ÐµÐ¼Ð²Ð¾Ð³Ð¹. Ð Ñ Ð°Ð²Ð¸Ð¸Ð»Ð¾
Ñ Ð°Ð±Ð¾Ñ Ð°ÐµÑ Ð² Ñ Ð¾Ð¾Ñ Ð²ÐµÑ Ñ Ñ Ð²Ð¸Ð¸Ð¸ Ñ Ñ Ð°Ñ Ð¿Ð¸Ñ Ð°Ð½Ð¸Ð¼
"Working hours". Ð Ñ Ð¸ Ñ Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°Ð½Ð¸Ð¸ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° Ð²
Ð¶Ñ Ñ Ð½Ð°Ð» Ð±Ñ Ð´ÐµÑ Ð·Ð°Ð¿Ñ Ð°Ð½Ð° Ð¸Ð½Ñ Ð¾Ñ Ð¼Ð°Ñ Ð¸Ñ
Ð½Ð°Ñ Ð°Ð»Ðµ Ñ ÐµÑ Ñ Ð¸Ð¸Ð».

Ð Ð¿Ð¾Ñ Ð»ÐµÐ´Ñ Ñ Ñ Ð¸Ñ Ñ Ñ Ð°Ñ Ñ Ñ Ñ Ð´Ð°Ð½Ð½Ð¾Ð³Ð¾ Ñ Ð°Ð·Ð´ÐµÐ»Ð°
Ð¼Ð¾Ð¶Ð½Ð¾ Ð½Ð°Ð¹Ñ Ð¸ Ð±Ð¾Ð»ÐµÐµ Ð¿Ð¾Ð´Ñ Ð¾Ð±Ð½Ñ Ð¹ Ð¿Ñ Ð¸Ð¼Ðµ Ñ Ð¿Ð¸Ñ Ð°Ð½Ð¸Ñ
Ð¿Ñ Ð°Ð²Ð¸Ð» UPL, Ð¾Ð¿Ð¸ÑÐ°Ð½Ð½Ð¾Ð³Ð¾ Ñ Ñ Ñ , Ð²Ñ Ñ Ñ Ð¾Ð³Ð¾Ð½Ð¾Ð¼ÐµÑ Ñ Ñ
Ð±Ð¾Ð±Ð¾Ñ Ð¾Ð¼ÐµÐºÐ°Ñ , Ñ Ñ Ð»Ð¾Ð²Ð¾Ñ Ñ Ñ , Ð´ÐµÐ¹Ñ Ñ Ð²Ñ Ñ Ñ ,
Ñ Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°Ñ Ð¸ Ñ Ð¸Ð¿Ð°Ñ Ð¿Ñ Ð°Ð²Ð¸Ð».

Р РµС РґР Р¿Р¾Р»Р¾Р¶РµРЅРёР¯ РґР

Product: NGFW
Version: 7.x

Р Р¾Р¼Р¼РµРЅС Р°Ñ Р¸Р

Р Ñ Р±Р°Ñ Ñ Ñ Ñ Р¾РºР°, Р½Р°Ñ Р¸Р½Р°Ñ Ñ Р°Ñ Ñ Ñ Ñ Ñ Р¸Р¼Р²Р¾Р»Р° "**%**", Ñ Р²Р»Ñ РµÑ Ñ Ñ ÐºÐ¾Р¼Р¼Р¼РµÐ½Ñ Р°Ñ Р¸РµÐ¼.
РiР¸Р¼Р²Р¾Р» Р¿Ñ Р¾Ñ РµÐ½Ñ Р° "**%**" Р¿Р¾Ñ Р»Рµ Р¿Ñ Р¾Р±РµÐ»Р° Р¸Р»Р¸ Ñ Р°Р±Ñ Р»Ñ Ñ Р¸Р¸ Р²Р²Р¾Р´Р¸Ñ ÐºÐ¾Р¼Р¼Р¼РµÐ½Ñ Р°Ñ Р¸Р¹, ÐºÐ¾Ñ Р¾Ñ Ð¹ Р¿Ñ Р¾Р´Р¾Р»Р¶Р°ÐµÑ Ñ Ñ Р´Р¾ ÐºÐ¾Р¼Р¼Р¼РµÐ½Ñ Р° Ñ Ñ Ñ Р¾ÐºÐ¸ (ÐºÑ Р¾Р¼Р¼Рµ Ñ Р»Ñ Ñ Р°ÐµÐ², ÐºÐ¾Р³Р´Р° Ñ Р¸Р¼Р²Р²Р¾Р» Р¿Ñ Р¾Ñ ÐµÐ½Ñ Р° Р¾Ñ Р¾Р±Ñ Р°Р¶Р°ÐµÑ Ñ Ñ Р²Ñ Ñ Ñ Р¸ ÐºÐ°Р²Ñ Ñ ÐµÐº ("), ÐºÐ°Ðº Ñ Р°Ñ Ñ Ñ Р²Ñ Ñ Р°Р¶ÐµÐ½Ð¸Ñ).

Р Ñ Р¸Р¼ÐµÑ :

```
% Р–Ñ Р¾ ÐºÐ¾Р¼Р¼Р¼ÐµÐ½Ñ Р°Ñ Р¸Р¹
DENY("Too many Host headers") request.header.Host.count = 2..  % Р¸ Ñ Ñ Р¾ Ñ Р¾Р¶Ðµ
```

Р Р¾Р¼Р¼ÐµÐ½Ñ Р°Ñ Р¸Р¸ Р¼Р¾Ñ Ñ Р±Ñ Ñ Ñ Р² Р»Ñ Р±Р¾Р¼ Р¼ÐµÑ Ñ Ðµ Ñ Р°Р¹Р»Р° Ñ Р¾Р¿Р¸Ñ Р°Р½ÐµÐ¼ Р¿Р¾Ñ Р¸Ñ ÐµÐ¼Ñ.

Р Ñ Р°Р²Р¸Р»Р°

Р Ñ Р°Р²Р¸Р»Р¾ Р¿Р¾Р»Р¸Ñ Р¸ÐºÐ¸ (rule) Ñ Р¾Ñ Ñ Р¾Р¸Ñ Р¸Р· Ñ Ñ Р»Р¾Р²Р¸Р¹ Р¸ Р½ÐµÐºÐ¾Ñ Р¾Ñ Р¾Р³Р¾ ÐºÐ¾Р»Р¸Ñ Ñ Р²Р° Р´ÐµÐ¹Ñ Ñ Р²Р¸Р¹, Р·Р°Р¿Ñ Р°Р½Ñ Ñ Р²Р° Р»Ñ Ð¿Ñ Р¾Ñ Ñ Р²Р° (properties), ÐºÐ¾Ñ Р¾Ñ Ñ Ðµ Ñ Р¸Ñ Р¾Ñ Р°ÐºÑ Р¸Ñ ÐµÑ ÐºÐ¸ Р²Ñ Р³Р»Ñ Р´Ñ Ñ ÐºÐ°Ðº Р´ÐµÐ¹Ñ Ñ Р²Р¸Рµ, Р½Р¾ Р¿Ñ Р¸ Ñ Ñ Р¾Р¼ Ð°ÐºÑ Р¸Р²Ñ Ñ Р´ÐµÐ¹Ñ Ñ Р²Р¸Р¹ Р½Ðµ Р¿Ñ Р¾Р¸Р·Р²Р¾Р´Ñ Ñ .
Р Р°Р¿Ñ Р¸Р¼ÐµÑ , Ñ Р²Р¾Р¹Ñ Ñ Р²Р¾ *name* Р¿Ñ Р¾Ñ Ñ Р¾ Р´Р¾Р±Р°Р²Р»Ñ ÐµÑ Р°Ñ Ñ Р¸Р±Ñ Ñ "Р¸Р¼Ñ " Р² Р¿Ñ Р°Р²Р¸Р»Р¾.

Р Ñ Р°Р²Р¸Р»Р° Р¾Р±Ñ Ñ Р½Р¾ Р¿Р¸Ñ Ñ Ñ Ñ Ñ Р² Р¾Р´Р½Р¾Р¹ Ñ Ñ Ñ Р¾ÐºÐµ, Р½Р¾ Р¼Р¾Р³Ñ Ñ Р±Ñ Ñ Ñ Р°Р·Р±Р¸Ñ Ñ Р½Р° Р¿Р¾Р¼Р¾Ñ Ñ Ñ Р¾Р¿ÐµÑ Р¸Р¸ Р¸Ñ Р¸Р¼Р¾Ñ Ñ Р¸Р¼Ñ Ð¿Ñ Р¾Ð¿Ñ Ñ Ñ Ñ Р¿ÐµÑ Р¸Р¸ Р¸Ñ Р¸Р¼Ñ Р¾Ñ Р»Ñ Р½Р¾Р³Р¾ Ñ Р¸Р¼Р²Р¾Р»Р° â Р¾Р±Ñ Р°Ñ Р½Р¾Р³Р¾ Ñ Р»ÐµÑ Р° "\".

Р Р¾Р³Р´Р° Р¿Ñ Р°Р²Р¸Р»Р¾
Р²Ñ Р¿Р¾Р»Р½Ñ ÐµÑ Ñ Ñ , Ñ Ñ Р»Р¾Р²Р¸Рµ Р¿Ñ Р¾Р²ÐµÑ Ñ ÐµÑ Ñ Ñ Р´Р»Ñ Ñ ÐµÐºÑ Ñ ÐµÐ¹ ÐºÐ¾Р¼Р¼Р½ÐºÑ ÐµÑ Р½Р¾Р¹ Ñ Ñ Р°Р½Р·Р°ÐºÑ Р¸Р¸.
Р Ñ Р»Р¸ Ñ Ñ Р»Р¾Р²Р¸Рµ Р¾Ñ ÐµÐ½Р¸Р²Р°ÐµÑ Ñ Ñ ÐºÐ°Ðº *True* (Р¸Ñ Ñ Р¸Р½Р°), Р²Ñ Р¿Р¾Р»Р½Ñ Ñ Ñ Ñ Ñ Р²Ñ Ðµ Ð¿ÐµÑ ÐµÑ Р¸Р»Р½Р½Ñ Ðµ Р´ÐµÐ¹Ñ Ñ Р²Р¸Ñ Р¸ Ñ ÐµÐºÑ Ñ Р¸Р¹ Ñ Р»Ð¾Р¹ Р·Р°Ð¿Ñ Р½Р°Ñ Ñ Ñ Р¿Ñ Р¾Р²Ð¾Ð´Ð¸Ñ ÐµÑ Р²Ð°ÐµÑ Ñ Ñ Р¿Ñ Р¾Ð²Ð¾Ð´Ð¸Ñ ÐµÑ Р² *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*. EcÐ»Ð¸ Ñ Ñ Р°Ð±Ñ Р°Р²Ñ ÐµÐµ Ð¿Ñ Р°Р²Ð¸Р»Ð° Р¾Ñ Р°Р²Ð¸Р»Ð° Р½Ðµ Ð¸ÐµÑ Ð¿Ñ Ð¾Ñ Ð¾Ñ Р² *PASS / FORCE_PASS / DENY / FORCE_DENY / WARNING / OK*, Ñ Ð¾ Р²Ñ Р¿Ñ Р¾Р»Ð½Ñ Ñ Ñ Ñ Ñ Р´ÐµÐ¹Ñ Ñ Р²Ð¸Ñ Ð¸ Р´Ð°Ð»Ñ Ñ Ðµ Ð¾Ñ Ð±Ñ Ð¾Ñ Ñ Р²Ð°ÐµÐ¼Ñ Ñ Ñ Ñ Ð¶Ðµ Ñ Ð»ÐµÐ´Ñ Ñ Ñ ÐµÐ¿Ñ Ð¿Ñ Р²Ð¸Ð¾. Р Ñ Ð»Ð¸ Ñ Ñ Ð»Ð¾Р²Ð¸Рµ Ð¾Ñ ÐµÐ½Ð¸Р²Ð°ÐµÑ Ñ Ñ ÐºÐ°Ðº *False* Р´Ð»Ñ Ñ Ñ Ð¾Ð¹ Ñ Ñ Ð°Ð½Ð·Ð°ÐºÑ Ð¸Ð¸, Ñ Ð¾ Р´Ð°Ð»Ñ Ñ Ðµ

Ð¾Ð±Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°ÐµÑ Ñ Ñ Ñ Ð¶Ðµ Ñ Ð»ÐµÐ´Ñ Ñ Ñ ÐµÐµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾Ð¼.

Ð Ñ Ðµ Ñ Ñ Ð»Ð¾Ð²Ð¸Ñ Ð² Ð¿Ñ Ð°Ð²Ð¸Ð»Ðµ Ð¿Ñ Ð¾Ð²ÐµÑ Ñ Ñ Ñ Ñ Ñ Ð¿Ð¾
Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¾Ð¼Ñ "*Ð* ". Ð Ñ Ñ Ð³Ð¸Ð¼Ð¸ Ñ Ð»Ð¾Ð²Ð°Ð°Ð¸Ð¸,
Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ñ Ñ Ð°Ð±Ð¾Ñ Ð°ÐµÑ , ÐºÐ¾Ð³Ð´Ð° Ð±Ñ Ð´Ñ Ñ
Ð²Ñ Ð¿Ð¾Ð»Ð½ÐµÐ½Ñ Ð²Ñ Ðµ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¸Ñ .

Ð Ñ Ð²Ð¾Ñ Ð¾Ñ ÐµÑ ÐµÐ´Ñ , Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¸Ðµ Ñ Ð²Ð»Ñ ÐµÑ Ñ Ñ
Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¾Ð¹ ÐºÐ¾Ð¼Ð¿Ð¾Ð¸Ð½Ð¾Ñ Ð¸ÐµÐ¹ Ñ Ñ Ð³Ð³Ð¸ÐµÑ Ð¾Ð².
Ð¢Ñ Ð¸Ð³Ð³ÐµÑ Ñ â Ñ Ñ Ð¾ Ð¾Ñ Ð´ÐµÐ»Ñ Ð½Ñ Ðµ Ñ ÐµÑ Ñ Ñ , ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ
Ð¼Ð¾Ð¶Ð¾Ð¾ Ð²Ñ Ð¿Ð¾Ð»Ð½Ð¸Ð¸Ñ Ñ Ñ ÐºÐ¾Ð¼Ð¿Ð¾Ð½Ð¾Ð²Ð½ÐµÐ½Ñ Ð°Ð¼Ð¸Ð¸
Ð·Ð°Ð¿Ñ Ð¾, Ð¾Ñ Ð²ÐµÑ Ð°, Ñ Ð²Ñ Ð·Ð°Ð½Ð½Ñ Ð¼Ð¸
Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ð¼Ð¸ Ð¸Ð»Ð¸ Ð¾Ð»Ð¸ Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸ÐµÐ¼
Ñ Ð¸Ñ Ñ ÐµÐ¼Ñ .

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ â Ñ Ñ Ð¾ Ð½Ð°Ñ Ñ Ñ Ð¾Ð¹Ð¾Ð¸, ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ
Ñ Ð¿Ñ Ð°Ð²Ð»Ñ Ñ Ñ Ð¾Ð±Ð°Ñ Ð°Ð±Ð¾Ñ Ñ Ñ Ð°Ð½Ð·Ð°Ð°ÐºÑ Ð¸Ð¸.
Ð Ð°Ð¿Ñ Ð¸Ð¼Ñ , Ð·Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ (*deny*) Ð¸Ð»Ð¸ Ð¾Ð±Ñ Ð±Ð°Ñ Ð°Ð±Ð¾Ñ Ñ
Ð¾Ð±Ñ ÐµÐºÑ (Ð¸Ð·Ð¼ÐµÐ½Ð¸Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº â *rewrite*).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Conditions ::= condition '\'? Conditions

Actions ::= action '\'? Actions

Ð Ñ Ð¸Ð¼ÐµÑ :

Ð Ð°Ð¿Ñ Ð¾Ñ Ð´ÐµÑ Ð·Ð°Ð¿Ñ ÐµÑ ÐµÐ½, ÐºÐ¾Ð³Ð´Ð° Ñ Ñ Ð°Ð±Ð¾Ñ Ð°Ñ
Ð¾Ð±Ð° Ñ Ñ Ð¸Ð³Ð³ÐµÑ Ð°:

- Ð´Ð¾Ð¼ÐµÐ½ Ð±Ñ Ð´ÐµÑ example.com
- Ð²Ñ ÐµÐ¼Ñ Ð±Ñ Ð´ÐµÑ Ð¼ÐµÐ¶Ð´Ñ 9 Ð¸ 17 Ñ Ð°Ñ Ð°Ð¼Ð¸

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

ÐiÐ»Ð¾Ð¸

ÐiÐ»Ð¾Ð¹ (layer)â Ñ Ñ Ð¾ ÐºÐ¾Ð½Ñ Ñ Ñ Ñ ÐºÑ Ð¸Ñ UPL,
Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ñ ÐµÐ¼Ð¾Ñ Ð´Ð»Ñ Ð³Ñ Ñ Ð¿Ð¿Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸
Ð¿Ñ Ð¸Ð½Ñ Ñ Ð¸Ñ Ð¾Ð´Ð½Ð¾Ð³Ð¾ Ñ ÐµÑ Ð¾Ð´Ð¾Ð¼ . Ð Ð°Ð·Ð´ÐµÐ»Ñ Ð½Ñ Ðµ
Ð¿Ñ Ð¸Ð½Ñ Ñ Ð¸Ñ Ñ ÐµÑ ÐµÐ½Ð¸Ñ Ð¿Ð¾Ð¼Ð¾Ð³Ð°Ñ Ñ
ÐºÐ¾Ð¼Ð¿Ñ Ñ Ð¾Ð»Ð¸Ñ Ð¾Ð²Ð°Ñ Ñ Ñ Ð»Ð¾Ð¶Ð¸Ð¼Ñ Ñ Ñ Ð¿Ð¾Ð¸Ñ Ð¸Ñ Ð¸Ð· Ð¾Ð´Ð¸Ð½
Ñ ÐµÑ ÐµÐ½Ð¸Ñ Ð² Ð¾Ñ Ð´ÐµÐ»Ñ Ð½Ð¾Ð¼.

Ð£ Ð»Ñ Ð±Ð¾Ð³Ð¾ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð² Ñ Ð»Ð¾Ðµ Ð¼Ð¾Ð¶Ð½Ð¾ Ð±Ñ Ð±Ñ
Ð¿Ñ ÐµÑ Ð¸ÐºÑ *PASS / FORCE_PASS / DENY / FORCE_DENY / OK / WARNING*, ÐºÐ¾Ð³Ð´Ð°
Ñ Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°ÐµÑ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ñ Ñ Ð°ÐºÑ Ð¾Ð¼ Ð¿Ñ ÐºÑ Ð¾Ð¼,
Ð²Ñ Ðµ
Ð¾Ñ Ñ Ð°Ð»Ñ Ðµ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° Ð² Ñ Ð»Ð¾Ðµ Ð¿Ð¾Ð¿Ð°Ð´Ð°Ñ Ñ Ñ Ð°ÐºÐ°Ñ Ñ Ñ Ñ .

Ð Ñ Ð»Ñ Ñ Ð°Ðµ ÐµÑ Ð»Ð¸ Ñ Ñ Ð°Ð±Ð¾Ñ Ð°Ð»Ð¾ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ñ
Ð¿Ñ ÐµÑ Ð¸ÐºÑ Ð¾Ð¼ *FORCE_PASS* Ð¸Ð»Ð¸ *FORCE_DENY*, Ñ Ð¾ Ñ Ñ Ð¾
Ñ Ð²Ð»Ñ ÐµÑ Ñ Ñ Ð¾ÐºÐ¾Ð½Ñ Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ñ Ð·Ð»Ñ Ð°Ñ Ð¾Ð¼
Ð¾Ð±Ñ Ð°Ð±Ð¾Ñ ÐºÐ¸, Ð² Ð¿Ñ Ð¾Ð²Ð½Ð¾Ð¼ Ñ Ð»Ñ Ñ Ð°Ðµ
Ð¾Ð±Ñ Ð°Ð±Ð¾Ñ ÐºÐ° Ð¿ÐµÑ ÐµÑ Ð¾Ð´Ð¸Ñ Ð½Ð° Ñ Ð»ÐµÐ´Ñ Ñ Ð¸Ð¹ Ñ Ð»Ð¾Ð¹.
Ð Ð¾Ñ Ð»Ðµ Ð¾Ð±Ñ Ð°Ð±Ð¾Ñ ÐºÐ¸ Ð²Ñ ÐµÑ Ñ Ð»Ð¾ÐµÐ² Ð·Ð°Ñ Ð¾Ð¿
Ð±Ñ Ð´ÐµÑ Ð·Ð°Ð»Ð¾Ð³Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ Ð¾Ð»Ð¸ Ð¿Ð¾Ð¿Ð°Ð´Ð°Ð½Ð¸Ñ Ñ ÐµÐ½ Ð²
Ð·Ð°Ð²Ð¸Ñ Ð¸Ð¼Ð¾Ñ Ñ Ð¸ Ð¾Ñ Ñ Ð¾Ð³Ð¾, Ñ Ñ Ð¾ Ð±Ñ Ð»Ð¾
Ð¿Ð¾Ð»Ð½Ð¸Ð¹ â *PASS / FORCE_PASS* Ð¸Ð»Ð¸ *DENY / FORCE_DENY*. Ð Ñ Ð»Ð¸
Ð¿Ñ Ð¾Ñ ÐµÑ Ñ Ð¸Ð½ Ð¾Ñ Ñ Ð°Ð½Ð¾Ð²Ð¸Ð¸Ñ Ñ Ñ Ð½Ð° *WARNING*, Ð±Ñ Ð´ÐµÑ
Ð´Ð¾Ð±Ð°Ð²Ð»ÐµÐ½Ð¾ Ð¿Ñ ÐµÐ´Ñ Ð¿Ñ ÐµÐ¶Ð´ÐµÐ½Ð¸Ðµ Ð² Ñ ÐµÐ»Ð¾
Ð¾Ñ Ð²ÐµÑ Ð°.

Ð Ñ ÐµÑ Ð¸ÐºÑ *OK* Ð¿Ð¾Ð´Ñ Ð°Ð·Ñ Ð¼ÐµÐ²Ð°ÐµÑ Ð¾Ñ Ñ Ð°Ð½Ð¾Ð²Ð¸ÐºÑ
Ð¾Ð±Ñ Ð°Ð±Ð¾Ñ ÐºÐ¸ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð² Ñ ÐµÐºÑ Ñ ÐµÐ¼ Ñ Ð»Ð¾Ðµ Ð¿Ñ Ð¸
Ð²Ñ Ð¿Ð¾Ð»Ð½ÐµÐ½Ð¸Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ð¸ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ð¹ (ÐµÑ Ð»Ð¸
Ñ Ð°ÐºÐ¾Ð²Ð¾Ñ Ðµ Ñ ÐºÐ°Ð·Ð°Ð½Ð¾). Ð Ñ Ð»Ð¸ Ð¿Ñ Ð¸ÐºÑ
Ð¾Ñ Ñ Ñ Ñ Ñ Ð²Ñ ÐµÑ Ð¿Ñ Ð¸Ð²Ð¾Ð»Ð¾Ð¶Ð½Ð¾ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ð¸
Ð´ÐµÐ¹Ñ Ð²Ð¸Ð¹, Ñ Ð¾ Ð¾Ñ Ñ Ð°Ð½Ð¾Ð²Ð¸Ð¸Ñ Ð½Ðµ
Ð¿Ð¾Ð´Ñ Ð°Ð·Ð½ÐµÐ²Ð°ÐµÑ Ñ Ñ .

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ *FORCE_PASS* Ð¸ *FORCE_DENY* Ð¿Ð¾Ñ Ð¾Ð¶Ð¸ Ð½Ð° *PASS* Ð¸ *DENY*,
Ð·Ð° Ð¸Ñ ÐºÐ»Ñ Ñ ÐµÐ½ÐµÐ¼ Ñ Ð¾Ð³Ð¾, Ñ Ñ Ð¾ Ð¾Ð½Ð¸ Ð¼Ð¾Ð³Ñ Ñ Ð¾Ð³Ð¾Ñ Ñ Ð±Ñ
Ð¿ÐµÑ ÐµÐ¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ñ Ð½Ð° Ð¿Ð¾Ð»Ð½Ð¾Ð¼ Ñ Ð¾Ð³Ð»Ð°ÑÐ¸Ð¸
Ñ Ð»Ð¾Ð²Ð¸Ñ Ñ . *FORCE_DENY* Ð¸ *FORCE_PASS* Ð½ÐµÐ¼Ð¾ÐµÐ´Ð»ÐµÐ½Ð½Ð¾
Ð¿Ñ ÐµÐºÑ Ð°Ñ Ð°Ñ Ñ Ð¿Ð¾Ð¾Ñ Ð¿Ñ Ð²Ð»Ñ ÐºÐ° Ð½Ð°Ð° Ð½Ð°
Ñ ÐµÐºÑ Ñ ÐµÐ¼, Ñ Ð°Ðº Ð¸ Ð½Ð° Ð¿Ñ Ð¾Ñ Ñ Ð²Ð»Ñ Ñ Ñ Ñ Ð¸Ñ Ñ Ð»Ð¾Ñ , Ð¸
Ñ Ñ Ð¾Ñ Ñ ÐµÐ·Ð·Ð»Ð¾Ñ Ð»Ñ Ñ Ð°Ñ Ñ Ð²Ð»Ñ ÐµÑ Ñ Ñ
Ð¾ÐºÐ¾Ð½Ñ Ð°Ñ ÐµÐ»Ñ Ð¼Ñ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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 ::= '"' Ð¿Ñ Ð¾Ð¸Ð·Ð²Ð¾Ð»Ñ Ñ Ð½Ð°Ñ Ñ Ñ Ð¾ÐºÐ° '"'

Ð Ñ Ð¸Ñ ÐµÑ 1:

```
[content "L1"]
DENY enabled(true) % Ð¿Ð¾ Ñ Ð¼Ð¾Ð»Ñ Ñ Ð°Ð½Ð¸Ñ Ð²Ñ Ðµ Ð·Ð°Ñ ÐµÑ ÐµÐ½Ð¸Ñ

[content "Devs"]
DENY group != Developers enabled(true)
%... Ð´Ð°Ð»Ñ Ñ Ðµ Ð¸Ð´Ñ Ñ Ð¿Ñ Ð²Ð°Ð»Ð°, ÐºÐ¾Ð¾Ñ Ñ Ðµ Ñ Ð±Ñ Ñ Ð¿Ñ Ð¸Ð¼ÐµÐ½Ñ Ñ Ñ Ñ Ñ Ð¾Ð»
```

Ð Ñ Ð¸Ñ ÐµÑ 2:

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

```
[content "L2"]
DENY enabled(true) % Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸ Đ²Ñ Đµ Đ·Đ°Đ¿Ñ ĐµÑ ĐµĐ½Đ¾
```

Đ Đ¸Đ½Đ°Đ¼Đ¸Ñ ĐµÑ ĐºĐ¸Đµ Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ

Đ Đ½Đ°Ñ ĐµĐ½Đ¸Ñ "Đ°Đ´Ñ ĐµÑ Đ·Đ°Đ¿Ñ Đ¾Ñ Đ°" *(url, url.host, url.path)*, "IP-Đ°Đ´Ñ ĐµÑ Đ¸Ñ Ñ Đ¾Ñ Đ½Đ¸ĐºĐ°/Đ½Đ°Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ " (*src.ip, dst.ip*), "Đ¸Đ¼Ñ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ " (*user*), "Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ Đ·Đ°Đ³Đ¾Đ»Đ¾Đ²ĐºÐ¾Đ²" (*request* Đ¸ *response*) Đ¸ "Đ¿Đ°Ñ Đ°Đ¼ĐµÑ Ñ Ñ Đ·Đ°Đ¿Ñ Đ¾Ñ Đ°" (*qparam*) Đ¼Đ¾Đ³Ñ Ñ Ñ Ñ Đ°Đ²Đ½Đ¸Đ²Đ°Ñ Ñ Ñ Ñ Đ¼ĐµĐ¶Đ´Ñ Ñ Đ¾Đ±Đ¾Đ¹, Đ° Ñ Đ°ĐºÐ¶Đµ Đ¸Ñ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ Ñ Ñ Ñ Đ² ĐºĐ°Ñ ĐµÑ Ñ Đ²Đµ Đ°Ñ Đ³Ñ Đ¼ĐµĐ½Ñ Đ° Đ² Đ´ĐµÐ¹Ñ Ñ Đ²Đ¸Ñ Ñ (actions), Đ³Đ´Đµ Ñ Ñ Đ¾ Đ¿Ñ ĐµÐ´Ñ Ñ Đ¼Đ¾Ñ Ñ ĐµÐ½Đ¾.

ÐŁÑ Đ»Đ¾Đ²Đ¸Ñ

Version: 7.x
Product: NGFW
Đ£Ñ Đ»Đ¾Đ²Đ¸Đµ (condition) Đ² Ñ Đ·Ñ ĐºĐµ UPL Ñ Đ²Đ»Ñ ĐµÑ Ñ Ñ Đ»Đ¾Đ³Đ¸Ñ ĐµÑ ĐºĐ¾Đ¹
ĐºĐ¾Đ¼Đ±Đ¸Đ½Đ°Ñ Đ¸ĐµÐ¹ Ñ Ñ Đ¸Đ³Đ³ĐµÑ Đ¾Đ². Đ¢Ñ Đ¸Đ³Đ³ĐµÑ Ñ â Ñ Ñ Đ¾Đ¾Ñ Đ´ĐµÐ»Ñ Đ½Ñ Đµ Ñ ĐµÑ Ñ Ñ , ĐºĐ¾Đ¾Ñ Đ¾Ñ Ñ Đµ Đ¼Đ¾Đ¶Đ½Đ¾ Đ²Ñ Đ¿Đ¾Đ»Đ½Đ¸Ñ Ñ Ñ ĐºĐ¾Đ¼Đ¿Đ¾Đ½ĐµÐ½Ñ Đ°Đ¼Đ¸ Đ·Đ°Đ¿Ñ Đ¾Ñ Đ°, Đ¾Ñ Đ²ĐµÑ Đ°, Ñ Đ²Ñ Đ·Đ°Đ½Đ½Ñ Đ¼Đ¸ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµÐ»Ñ Đ¼Đ¸ Đ¸Đ»Đ¸ Ñ Đ¾Ñ Ñ Đ¾Ñ Đ½Đ¸ĐµÐ¼ Ñ Đ¸Ñ Ñ ĐµÐ¼Ñ . Đ Ñ Đµ Ñ Ñ Đ¸Đ³Đ³ĐµÑ Ñ Ñ Ñ Đ»Đ¾Đ²Đ¸Ñ Ñ Ñ Đ°Đ²Đ½Đ¸Đ²Đ°Ñ Ñ Ñ Ñ Ñ Đ¾ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Ñ Đ¼Đ¸ Đ¿Đ¾Đ¼Đ¾Ñ Ñ Ñ Đ¾Đ¿ĐµÑ Đ°Ñ Đ¾Ñ Đ¾Đ² "=" Đ¸ "!=". Đ Ñ Đ¾Đ»Đ¸ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Ñ Đ¼Đ¾Đ³Ñ Ñ Đ²Ñ Ñ Ñ Ñ Đ¿Đ°Ñ Ñ ĐºĐ¾Đ½Ñ Ñ Đ°Đ½Ñ Đ½Ñ Đµ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Ñ , Ñ Đ°ĐºĐ¸Đµ ĐºĐ°Đº Ñ Ñ Ñ Đ¾ĐºĐ¸, Ñ ĐµÐ»Đ¾Ñ Đ¸Ñ Đ»ĐµÐ½Đ½Ñ Đµ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Ñ Đ¸Đ¸ Đ´Đ¸Đ½Đ°Đ¼Đ¸Ñ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Ñ ĐµÐ½Đ¸Ñ , Đ´Đ¸Đ°Đ¿Đ°Đ·Đ¾Đ½Ñ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Đ¹, Đ´Đ¸Đ½Đ°Đ¼Đ¸Ñ ĐµÑ ĐºĐ¸Đµ Đ·Đ½Đ°Ñ ĐµÐ½Đ¸Ñ .

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

condition ::= condition_name ('=' | '!=') condition_value

condition_value ::= pattern | list

list ::= '(' ((pattern ',') pattern)? ')'*

pattern ::= word | string | integer | float | boolean| range | condition_name

string ::= '"' Đ¿Ñ Đ¾Đ¸Đ·Đ²Đ¾Đ»Ñ Đ½Đ°Ñ Ñ Ñ Ñ Đ¾ĐºĐ° '"'

*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

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð½Ð°Ð»Ð¸Ñ Ð¸Ñ *HTTP CONNECT.*

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

http.connect = yes | no | true | false

http.method

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ð¹ HTTP-Ð¼ÐµÑ Ð¾Ð´Ð°.
Ð ÐµÑ Ð¾Ð´ Ð¼Ð¾Ð¶Ð½Ð¾ Ñ ÐºÐ°Ð·Ñ Ð²Ð°Ñ Ñ ÐºÐ°Ðº Ð² ÐºÐ°Ð²Ñ Ñ ÐºÐ°Ñ , Ñ Ð°Ðº Ð¸ Ð±ÐµÐ·.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð²ÐµÑ Ñ Ð¸Ð¸ HTTP-Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

http.request.version = 0.9 | 1.0 | 1.1

http.response.version

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð²ÐµÑ Ñ Ð¸Ð¸ HTTP-Ð¾Ñ Ð²ÐµÑ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

http.response.version = 0.9 | 1.0 | 1.1

http.response.code

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° HTTP-ÐºÐ¾Ð´Ð° Ð¾Ñ Ð²ÐµÑ Ð°. Ð Ð°Ð»Ð¸Ð´Ð½Ñ Ðµ Ð·Ð½Ð°ÑÐµÐ½Ð¸Ñ : 100 - 999.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

http.response.code = NNN %(Ð³Ð´Ðµ NNN Ñ Ð¸Ñ Ð»Ð¾ Ð¾Ñ 100 Ð´Ð¾ 999)

http.request.body, http.request.body.nocase, http.response.body Ð¸ http.response.body.nocase

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ ÐµÐ»Ð° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð° HTTP Ð½Ð° Ñ Ð¾Ð´ÐµÑ Ð¶Ð°Ð½Ð¸Ðµ Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð½Ð¾Ð¹ Ñ Ð¸Ð³Ð½Ð°Ñ Ñ Ñ Ñ .

Ð Ñ Ð¸Ð¼ÐµÑ :

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

category

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð´Ð¾Ð¼ÐµÐ½Ð° Ð½Ð° Ð¿Ñ Ð¸Ð½Ð°Ð´Ð»ÐµÐ¶Ð½Ð¾Ñ Ñ Ñ Ðº Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð½Ð¾Ð¹ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¸ Ñ Ð°Ð¹Ñ Ð¾Ð² (Ñ Ð¿Ð¸Ñ Ð¾Ðº Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð½Ñ Ñ Ð² Ñ Ð¸Ñ Ñ ÐµÐ¼Ðµ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¹ Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ð² Ñ Ñ Ð°Ñ Ñ Ðµ Ð ÐºÑ ÐµÐ³Ð¾Ñ Ð¸Ð¸ URL).
Ð Ð°Ñ ÐµÐ³Ð¾Ñ Ð¸Ñ Ð¼Ð¾Ð¶Ð½Ð¾ Ñ ÐºÐ°Ð·Ð°Ñ Ñ ÐºÐ°Ðº Ð¿Ð¾ Ð¸Ð¼ÐµÐ½Ð¸, Ñ Ð°Ðº Ð¸ Ð¿Ð¾ ÐµÐµ Ð¸Ð´ÐµÐ½Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¾Ñ Ñ . Ð Ñ Ð»Ð¸ Ð¸Ð¼Ñ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¸ Ñ Ð¾Ð´ÐµÑ Ð¶Ð¸Ñ Ð¿Ñ Ð¾Ð±ÐµÐ»Ñ Ð¸Ð»Ð¸ Ñ Ð¿ÐµÑ Ñ Ð¸Ð¼Ð²Ð¾Ð»Ñ , ÐµÐµ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾ Ñ ÐºÐ°Ð·Ñ Ð²Ð°Ñ Ñ Ð² ÐºÐ°Ð²Ñ Ñ ÐºÐ°Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

category = word | string | integer | list | lib

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

list_libs :: = lib_name ','' list_libs

lib_name ::= word | string

Ð Ñ Ð¸Ð¼ÐµÑ :

Ð Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¸ *Job Search* Ð¸ *Gambling:*

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

Ð Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ Ð²Ñ Ðµ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¸ Ð¸Ð· Ð±Ð¸Ð±Ð»Ð¸Ð¾Ñ ÐµÐºÐ¸ *Restricted cats:*

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

morphology

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ ÐµÐ»Ð° Ð¾Ñ Ð²ÐµÑ Ð° Ð¼Ð¾Ñ Ñ Ð¾Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¸Ð¼Ð¸ Ñ Ð»Ð¾Ð²Ð°Ñ Ñ Ð¼Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

morphology = word | string | list | lib

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

list_libs :: = lib_name ','' list_libs

lib_name ::= word | string

Ð Ñ Ð¸Ð¼ÐµÑ :

Ð Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ ÐºÐ¾Ð½Ñ ÐµÐ½Ñ , ÐµÑ Ð»Ð¸ Ñ Ñ Ð°Ð±Ð¾Ñ Ð°ÐµÑ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ñ Ð¼Ð¾Ñ Ñ Ð¾Ð»Ð¾Ð³Ð¸Ð¸ Ð¸Ð· Ñ Ð»Ð¾Ð²Ð°Ñ Ñ *BadWords:*

```
DENY morphology = BadWords
```

Ð Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ Ð¼Ð¾Ñ Ñ Ð¾Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¸Ðµ ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¸ Ð¸Ð· Ð±Ð¸Ð±Ð»Ð¸Ð¾Ñ ÐµÐº **Special Words**, **BadWords**

```
DENY morphology = ("Special Words", BadWords)
DENY morphology = lib.morphology("Special Words", BadWords) % Ð°Ð½Ð°Ð»Ð¾Ð³Ð¸Ñ Ð½Ð¾ Ð¿Ñ ÐµÐ´Ñ Ð´Ñ Ñ
```

request.header.<h_name> Ð¸ response.header.<h_name>

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð°. *h_name* Ð¼Ð¾Ð¶ÐµÑ Ð¿Ñ Ð¸Ð½Ð¸Ð¼Ð°Ñ Ñ Ð¾Ð´Ð½Ð¾ Ð¸Ð· Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ð¹ (Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ñ Ñ Ð°Ñ Ñ Ñ ÐiÐ¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð²).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

```
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 Ð¸ response.header.<h_name>.substring

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð° Ð½Ð° Ð²Ñ Ð¾Ð¶Ð´ÐµÐ½Ð¸Ðµ Ð¿Ð¾Ð´Ñ Ñ Ñ Ð¾ÐºÐ¸. *h_name* Ð¼Ð¾Ð¶ÐµÑ Ð¿Ñ Ð¸Ð½Ð¸Ð¼Ð°Ñ Ñ Ð¾Ð´Ð½Ð¾ Ð¸Ð· Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ð¹ (Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ñ Ñ Ð°Ñ Ñ Ñ ÐiÐ¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð²).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

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

request.header.<h_name>.regex Ð¸ response.header.<h_name>.regex

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð° Ð½Ð° Ñ ÐµÐ³Ñ Ð»Ñ Ñ Ð½Ð¾Ðµ Ð²Ñ Ñ Ð°Ð¶ÐµÐ½Ð¸Ðµ PCRE. *h_name* Ð¼Ð¾Ð¶ÐµÑ

Ð¿Ñ Ð¸Ð½Ð¼Ð°Ñ Ñ Ð¾Ð´Ð½Ð¾ Ð¸Ð· Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð½Ð°Ñ Ð¹ (Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ñ Ñ Ð°Ñ Ñ Ñ ÐiÐ¿Ð¸Ñ Ð¾Ð¿ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐ¾Ð²).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

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

request.header.<h_name>.re2 Ð¸ response.header.<h_name>.re2

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐ° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð° Ð½Ð° Ñ ÐµÐ³Ñ Ð»Ñ Ñ Ð½Ð¾Ðµ Ð²Ñ Ñ Ð°Ð¶ÐµÐ½Ð¸Ðµ RE2. *h_name* Ð¼Ð¾Ð¶ÐµÑ Ð¿Ñ Ð¸Ð½Ð¸Ð¼Ð°Ñ Ñ Ð¾Ð´Ð½Ð¾ Ð¸Ð· Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð½Ð°Ñ Ð¹ (Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ñ Ñ Ð°Ñ Ñ Ñ ÐiÐ¿Ð¸Ñ Ð¾Ð¿ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐ¾Ð²).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

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

request.header.<h_name>.count Ð¸ response.header.<h_name>.count

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° ÐºÐ¾Ð»Ð¸ÑÐµ Ñ Ð²Ð° Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð² <*h_name*> Ð² HTTP-Ð·Ð°Ð¿Ñ Ð¾Ñ Ðµ/Ð¾Ñ Ð²ÐµÑ Ðµ. *h_name* Ð¼Ð¾Ð¶ÐµÑ Ð¿Ñ Ð¸Ð½Ð¸Ð¼Ð°Ñ Ñ Ð¾Ð´Ð½Ð¾ Ð¸Ð· Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð½Ð°Ñ Ð¹ (Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ñ Ñ Ð°Ñ Ñ Ñ ÐiÐ¿Ð¸Ñ Ð¾Ð¿ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐ¾Ð²).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

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

request.header.<h_name>.length Ð¸ response.header.<h_name>.length

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð´Ð»Ð¸Ð½Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ð²Ñ Ñ Ð¾Ð²Ð¾ÐºÐ° <*h_name*> Ð² HTTP-Ð·Ð°Ð¿Ñ Ð¾Ñ Ðµ/Ð¾Ñ Ð²ÐµÑ Ðµ. *h_name* Ð¼Ð¾Ð¶ÐµÑ Ð¿Ñ Ð¸Ð½Ð¸Ð¼Ð°Ñ Ñ Ð¾Ð´Ð½Ð¾ Ð¸Ð· Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð½Ð°Ñ Ð¹ (Ñ Ð¼Ð¾Ñ Ñ Ð¸Ñ Ðµ Ñ Ñ Ð°Ñ Ñ Ñ ÐiÐ¿Ð¸Ñ Ð¾Ð¿ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐ¾Ð²).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

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

request.header_names, request.header_values, response.header_values Ð¸ response.header_values

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð¸Ð¼ÐµÐ½Ð¸/Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ð²Ñ ÐµÑ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð² Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð° Ð½Ð° Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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> Ð¸ response.x_header.<xh_name>

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°/Ð¾Ñ Ð²ÐµÑ Ð° Ð½Ð° Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ. *xh_name* â Ð¿Ñ Ð¾Ð¸Ð·Ð²Ð¾Ð»Ñ Ñ Ð¹ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

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

Ð Ð¾Ð·Ð¼Ð¾Ð¶Ð½Ñ Ñ Ð°ÐºÐ¶Ðµ Ñ Ñ Ñ Ð¸ÐºÑ Ñ **length**, **count**, **regex**, **re2** ÐºÐ°Ðº Ð¸ Ð² Ñ Ð»Ñ Ñ Ð°Ðµ Ñ *<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>

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ° Ð·Ð°Ð¿Ñ Ð¾Ñ Ð° Cookie Ð½Ð° Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

```
DENY http.method = POST request.header.Cookie.csrf_token != qparam.CSRF_TOKEN enabled(true) name('
```

time, day, hour, minute

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ð¾Ñ Ð²ÐµÑ Ñ Ñ Ð²Ð¸Ñ Ñ ÐµÐºÑ Ñ ÐµÐ³Ð¾ Ð²Ñ Ð¼Ñ ÐµÐ½Ð¸ Ð·Ð°Ð´Ð°Ð½Ð½Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ñ Ð¸Ð½ÑÐµÑ Ñ Ð¸Ñ Ð¾Ñ Ð¸Ñ *utc*, Ð²Ñ ÐµÐ¼Ñ Ð±ÐµÑ ÐµÐ½Ñ Ñ Ñ Ð»Ð¾ÐºÐ°Ð»Ñ Ð½Ð¾Ð¹ â Ð¿Ð¾ Ð Ñ Ð¸Ð½Ð²Ð¸Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

```
PASS time = 12:00..13:00  % Ñ Ð°Ð·Ñ ÐµÑ Ð¸Ñ Ñ  ÐºÐ°Ð¶Ð´Ñ Ð¹ Ð´ÐµÐ½Ñ  Ñ  12 Ð´Ð¾ 13 Ñ Ð°Ñ Ð¾Ð²
PASS time = lib.time("Ð Ñ Ð°Ð·Ð´Ð½Ð¸Ðº") % Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ñ Ñ  Ð±Ð¸Ð±Ð»Ð¸Ð¾ÑÐµÐºÐµ  "Ð Ñ Ð°
DENY day = (sunday, saturday) % Ð·Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ  Ð½Ð° Ð²Ñ Ñ Ð¾Ð´Ð½Ñ Ñ
DENY day = (monday, 15) hour = 9..18  % Ð·Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ  ÐºÐ°Ð¶Ð´Ñ Ð¹ Ð¿Ð¾Ð½ÐµÐ´ÐµÐ»ÑÐ½Ð¸Ðº Ð½Ð¸Ð¼ Ð¸
```

Ð Ñ ÐºÑ Ñ Ñ Ñ Ðµ Ð¸Ð½Ñ ÐµÑ Ð²Ð°Ð» Ñ Ñ Ð¸Ñ Ñ Ð²Ð°Ñ Ñ Ñ Ñ Ð¿Ð¾ Ð³Ñ Ð°Ð½Ð¸ Ñ Ñ Ñ Ð¾Ð²/Ñ Ð°Ñ Ð°.

```
PASS hour = 18.. % Ð¾Ð·Ð½Ð°Ñ Ð°ÐµÑ , Ñ Ñ Ð¾ Ñ Ð°Ð·Ñ ÐµÐ½Ð¾Ñ  18 Ñ Ð°Ñ Ð¾Ð² Ð´Ð¾ Ð¿Ð¾Ð»Ñ Ð½Ð¾Ñ
minute = ..10 % Ð¿ÐµÑ Ð²Ñ Ðµ 10 Ð¼Ð¸Ð½ Ñ  ÐºÐ°Ð¶Ð´Ð¾Ð¾Ð¾ Ñ Ð°Ñ Ð°
```

url, url.host Ð¸ url.address

Ð ÑÐ¾Ð²ÐµÑÐºÐ° url Ð¸Ð»Ð¸ ÐµÐ³Ð¾ ÑÐ°ÑÑÐ¸ÑÐµÐ½Ð¸Ñ Ð·Ð½Ð°ÑÐµÐ½Ð¸Ñ Ð·Ð°Ð´Ð°ÑÑ Ð¾Ð±ÑÐ°Ð·ÐµÑ ÐµÐ¾ÑÐ¸ÑÐ°ÑÐµÐ»Ñ Ð¸Ð»Ð¸ ÐµÐ³Ð¾ Ð¸ÑÐ¿Ð¾Ð»Ñ Ð·ÑÐµÑÑÑ Ð¿Ð¾Ð»Ð½ÑÐ¹ URI c Ð´ÐµÐºÐ¾Ð´Ð¸ÑÐ¾Ð²Ð°Ð½Ð¸ÐµÐ¼ *%*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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 % Ð¿Ð¾Ð»Ð½Ñ Ð¹ Ð¸Ð»Ð¸ Ð½ÐµÑ URL

prefix ::= string % Ð½Ð°Ñ Ð°Ð»Ð¾ Ñ Ñ Ñ Ð¾ÐºÐ¸

substring ::= string % Ð¿Ð¾Ð´Ñ Ñ Ñ Ð¾ÐºÐ°

suffix ::= string % Ð¾ÐºÐ¾Ð½Ñ Ð°Ð½Ð¸Ðµ Ñ Ñ Ñ Ð¾ÐºÐ¸

regex ::= string % Ñ ÐµÐ³Ñ Ð»Ñ Ñ Ð½Ð¾Ðµ Ð²Ñ Ñ Ð°Ð¶ÐµÐ½Ð¸Ðµ PCRE

re2 ::= string % Ñ ÐµÐ³Ñ Ð»Ñ Ñ Ð½Ð¾Ðµ Ð²Ñ Ñ Ð°Ð¶ÐµÐ½Ð¸Ðµ RE2

url.address â Ñ Ñ Ð¾, Ð¿Ð¾ Ñ Ñ Ñ Ð¸, Ñ Ð¸Ð½Ð¾Ð½Ð¸Ð¼ *dst.ip*.

Ð Ñ Ð¸Ð¼ÐµÑ :

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

qparam.<name>, qparam.values Ð¸ qparam.names

Ð ÑÐ¾Ð²ÐµÑÐºÐ° Ð·Ð½Ð°ÑÐµÐ½Ð¸Ñ Ð¿Ð°ÑÐ°Ð¼ÐµÑÑ Ð¾Ð´Ð½Ð¾Ð³Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÐ° Ð¸Ð»Ð¸ ÐµÐ³Ð¾ Ð¸ÑÐ¿Ð¾Ð»Ñ Ð·ÑÐµÑÑÑ Ð¿Ð¾Ð»Ð½ÑÐ¹ Ð¸Ð¼ÐµÐ½Ð¸ Ð¿Ð°ÑÐ°Ð¼ÐµÑÐ° c Ð´ÐµÐºÐ¾Ð´Ð¸ÑÐ¾Ð²Ð°Ð½Ð¸ÐµÐ¼ *%*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

qparam.length = numeric % Ð¿Ñ Ð¾Ð²ÐµÑÐ¸ Ñ Ñ Ð¾Ð±Ñ Ñ Ñ Ð´Ð»Ð¸Ð½Ñ query-Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¾Ð²

qparam.count = numeric % Ð¿Ñ Ð¾Ð²ÐµÑÐ¸ Ñ Ñ ÐºÐ¾Ð»Ð¸ÑÐµÑÑÐ²Ð¾ query-Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¾Ð² qparam.<name>[.(length | count)] = numeric

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

qparam.values[.base64].substring[.nocase] = string % Đ¿Ñ Đ¾Đ²ĐµÑ Đ¸Ñ Ñ Đ²Ñ Đµ Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ Đ½Đ° Đ²Ñ Đ¾Đ¶Đ´ĐµĐ½Đ¸Đµ Đ¿Đ¾Đ´Ñ Ñ Ñ Đ¾ĐºĐ¸

qparam.names[.base64].substring[.nocase] = string % Đ¿Ñ Đ¾Đ²ĐµÑ Đ¸Ñ Ñ Đ²Ñ Đµ Đ¸Đ¼ĐµĐ½Đ° Đ½Đ° Đ²Ñ Đ¾Đ¶Đ´ĐµĐ½Đ¸Đµ Đ¿Đ¾Đ´Ñ Ñ Ñ Đ¾ĐºĐ¸

qparam.values[.base64].regex = string % Đ¿Ñ Đ¾Đ²ĐµÑ Đ¸Ñ Ñ Đ²Ñ Đµ Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ Đ½Đ° Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Đ¾Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Đµ

qparam.names[.base64].regex = string % Đ¿Ñ Đ¾Đ²ĐµÑ Đ¸Ñ Ñ Đ²Ñ Đµ Đ¸Đ¼ĐµĐ½Đ° Đ½Đ° Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Ñ Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Ñ

qparam.values[.base64].re2 = string % Đ¿Ñ Đ¾Đ²ĐµÑ Đ¸Ñ Ñ Đ²Ñ Đµ Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ Đ½Đ° Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Đ¾Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Đµ

qparam.names[.base64].re2 = string % Đ¿Ñ Đ¾Đ²ĐµÑ Đ¸Ñ Ñ Đ²Ñ Đµ Đ¸Đ¼ĐµĐ½Đ° Đ½Đ° Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Ñ Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Ñ numeric ::= integer | range % Ñ Đ¸Ñ Đ»Đ¾ Đ»Đ¸Đ±Đ¾ Đ´Đ¸Đ°Đ¿Đ°Đ·Đ¾Đ½

regex ::= string % Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Đ¾Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Đµ PCRE

re2 ::= string % Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Đ¾Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Đµ RE2

Đ Ñ Đ¸Đ¼ĐµÑ :

```
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 Đ¸ group

Đ Ñ Đ¾Đ²ĐµÑ ĐºĐ° Ñ ĐµĐºÑ Ñ ĐµĐ³Đ¾ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ Đ¸Đ»Đ¸ ĐµĐ³Đ¾ Đ³Ñ Ñ Đ¿Đ¿Ñ .

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

user = word | string | known | unknown

group = word | string

user.guid = string

group.guid = string

known â Đ¸Ñ Đ¿Đ¾Đ»Ñ Đ·Ñ ĐµÑ Ñ Ñ Đ´Đ»Ñ Ñ ĐºĐ°Đ·Đ°Đ½Đ¸Ñ Đ°Đ²Ñ Đ¾Ñ Đ¸Đ·Đ¾Đ²Đ°Đ½Đ½Đ¾Đ³Đ¾ (Đ¸Đ·Đ²ĐµÑ Ñ Đ½Đ¾Đ³Đ¾) Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ ;
unknown â Đ¸Ñ Đ¿Đ¾Đ»Ñ Đ·Ñ ĐµÑ Ñ Ñ Đ´Đ»Ñ Ñ ĐºĐ°Đ·Đ°Đ½Đ¸Ñ Đ½ĐµĐ°Đ²Ñ Đ¾Ñ Đ¸Đ·Đ¾Đ²Đ°Đ½Đ½Đ¾Đ³Đ¾ (Đ½ĐµĐ¸Đ·Đ²ĐµÑ Ñ Đ½Đ¾Đ³Đ¾) Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ .

Đ Ñ Đ¸Đ¼ĐµÑ :

```
PASS user = known % Ñ Đ°Đ·Ñ ĐµÑ Đ¸Ñ Ñ  Đ¸Đ·Đ²ĐµÑ Ñ Đ½Ñ Ñ  Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»ĐµĐ¹
```

```
DENY group = "Ð Ñ Ð´ÐµÐ» Ð¿Ñ Ð¾Ð´Ð°Ð¶" category = "Pornography/Sexually Explicit"   % Ð·Ð°Ð¿Ñ ÐµÑ
```

src Ð¸ dst

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ñ Ð»Ð¾Ð²Ð¸Ñ Ð½Ð° IP-Ð°Ð´Ñ ÐµÑ , Ð·Ð¾Ð½Ñ Ð¸Ð» GeoIP Ð¸Ñ Ñ Ð¾Ñ Ð½Ð¸ÐºÐ°/Ð½Ð°Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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 â Ñ Ñ Ð¾, Ð¿Ð¾ Ñ Ñ Ñ Ð¸, Ñ Ð¸Ð½Ð¾Ð½Ð¸Ð¼ *dst.ip*.

scenario

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð°ÐºÑ Ð¸Ð²Ð½Ð¾Ñ Ñ Ð¸ Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð½Ð¾Ð³Ð¾ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

scenario = string | word | list

Ð Ñ Ð¸Ð¼ÐµÑ :

```
DENY scenario = Torrents  desc("Ð °Ð¿Ñ ÐµÑ Ð¸Ñ Ñ , ÐµÑ Ð»Ð¸ Ð°ÐºÑ Ð¸Ð²ÐµÐ½ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ð¹ Torre
```

virus_heuristic Ð¸ virus_usergate

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ ÐµÐ»Ð° Ð¾Ñ Ð²ÐµÑ Ð° Ð½Ð° Ð²Ð¸Ñ Ñ Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

virus_heuristic = yes | no | true | false % (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ false)

virus_usergate = yes | no | true | false % (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ false)

heuristic â Ñ Đ²Ñ Đ¸Ñ Ñ Đ¸Ñ ĐµÑ ĐºĐ¸Đ¹ Đ°Đ½Đ°Đ»Đ¸Đ·Đ°Ñ Đ¾Ñ (Đ¼ĐµĐ´Đ»ĐµĐ½Đ½Ñ Đ¹);
usergate â Đ¿Ñ Đ¾Đ²ĐµÑ ĐºĐ° Đ¿Đ¾ Ñ Ñ Ñ Đ°Đ¼ (Đ±Ñ Ñ Ñ Ñ Ñ Đ¹).

bridge_vlan_filter

Đ¤Đ¸Đ»Ñ Ñ Ñ Đ°Ñ Đ¸Ñ Ñ Ñ Đ°Ñ Đ¸ĐºĐ° Đ¿Đ¾ VLAN-Ñ ĐµĐ³Đ°Đ¼ Đ´Đ»Ñ Đ¸Đ½Ñ ĐµÑ Ñ ĐµĐ¹Ñ Đ° Đ² Ñ ĐµĐ¶Đ¸Đ¼Đµ Đ¼Đ¾Ñ Ñ Đ° (bridge).

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

Đ Ñ Đ¸Đ¼ĐµÑ :

```
DENY bridge_vlan_filter = (10, 100..200)  desc("Đ Đ°Đ¿Ñ ĐµÑ Đ¸Ñ Ñ Ñ Đ°Ñ Đ¸Đº Đ¿Đ¾ VLAN-Ñ ĐµĐ³Đ°
```

service

Đ ĐµÑ ĐµĐºÑ Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Đµ Ñ Ñ Đ°Ñ Đ¸ĐºĐ° Đ¾Đ¿Ñ ĐµĐ´ĐµĐ»ĐµĐ½Đ½Đ¾Đ³Đ¾ Ñ ĐµÑ Đ²Đ¸Đ° Đ´Đ»Ñ Đ¼ĐµĐ¶Ñ ĐµÑ ĐµĐ²Đ¾Đ³Đ¾ Ñ ĐºÑ Đ°Đ½Đ°.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

service = string | word | list

Đ Ñ Đ¸Đ¼ĐµÑ :

```
DENY service = POP3  desc("Đ Đ°Đ¿Ñ ĐµÑ Đ¸Ñ Ñ ĐµÑ Đ²Đ¸Ñ POP3")
```

application

Đ ĐµÑ ĐµĐºÑ Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Đµ Ñ Ñ Đ°Ñ Đ¸ĐºĐ° Đ¾Đ¿Ñ ĐµĐ´ĐµĐ»ĐµĐ½Đ½Đ¾Đ³Đ¾ Đ¿Ñ Đ¸Đ»Đ¾Đ¶ĐµĐ½Đ¸Ñ Đ½Đ° Ñ Ñ Đ¾Đ²Đ½Đµ L7 Đ´Đ»Ñ Đ¼ĐµĐ¶Ñ ĐµÑ ĐµĐ²Đ¾Đ³Đ¾ Ñ ĐºÑ Đ°Đ½Đ°.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

application = string | word | list

Đ Ñ Đ¸Đ¼ĐµÑ :

```
DENY application = Tor  desc("Đ Đ°Đ¿Ñ ĐµÑ Đ¸Ñ Ñ  Tor")
```

envelope_from Đ¸ envelope_to

Đ Ñ Đ¾Đ²ĐµÑ ĐºĐ° email-Đ°Đ´Ñ ĐµÑ Đ° Đ¾Ñ Đ¿Ñ Đ°Đ²Đ¸Ñ ĐµĐ»Ñ /Đ°Đ´Ñ ĐµÑ Đ°Ñ Ñ Đ¾Đ¾Đ±Ñ ĐµĐ½Đ¸Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

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

Ð Ñ Ð¸Ð¼ÐµÑ :

```
PASS envelope_from = "Email froup from" envelope_to = "Email froup to" service = SMTP mark_hdr(Sub
```

response_time

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð²Ñ ÐµÐ¼Ð½Ð¸ Ð¾Ñ Ð²ÐµÑ Ð° Ð²
Ð¼Ð¸Ð»Ð»Ð¸ ÐµÐºÑ Ð½Ð´Ð°Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

response_time = integer

hip_profile

HIP-Ð¿Ñ Ð¾Ñ Ð¸Ð»Ð¸ Ð´Ð»Ñ Ð¿Ð¾Ð²Ð¾ÐµÑ ÐºÐ¸ Ð¾Ñ Ð¾Ð¾Ñ Ð²ÐµÑ Ñ Ñ Ð²Ð¸Ñ
ÐºÐ¾Ð½ÐµÑ Ð½Ð¾Ð³Ð¾ Ñ Ñ Ñ Ñ Ð¾Ð¾Ð¹Ñ Ñ Ð²Ð° Ñ Ñ ÐµÐ±Ð¾Ð²Ð°Ð½Ð¸Ñ
Ð±ÐµÐ·Ð¾Ð¿Ð°Ñ Ð½Ð¾Ñ Ñ Ð¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾Ð¹ Ñ Ð¾Ð»Ñ ÐºÐ¾
Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÐ²Ð¾Ð¹ Ñ ÐºÑ Ð°Ð½Ð½Ñ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

hip_profile = string | word | list

Ð Ñ Ñ Ð¾ÐµÐ½Ð½Ñ Ðµ Ð±Ð¸Ð±Ð»Ð¸Ð¾ÐµÐºÐ

Version: 7.x
Product: NGFW
Ð Ð¸Ð±Ð»Ð¸Ð¾ÐµÐºÐ¸ (*lib*) â Ñ Ñ Ð¾ Ñ Ð»ÐµÐ¼ÐµÐ½Ñ Ñ Ð·Ñ ÐºÐ° UPL,
ÐºÐ¾Ð¾Ñ Ñ Ðµ Ñ Ð»Ñ Ð¶Ð°Ñ Ð´Ð»Ñ Ð´Ð¾Ñ Ñ Ñ Ð¿Ð° Ðº
Ð²Ñ Ñ Ñ Ð¾ÐµÐ½Ð½Ñ Ð¼ Ð¸ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ Ñ ÐºÐ¸Ð¼
Ð±Ð¸Ð±Ð»Ð¸Ð¾Ñ ÐµÐºÐ¼. Ð Ð°Ðº Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾, Ñ Ñ Ð¾Ð¾
Ð´Ð¾Ñ Ñ Ð°Ñ Ð¾Ñ Ð½Ð¾ Ð±Ð¾Ð»Ñ Ñ Ð¸Ð¸ Ñ Ð¿Ð¸Ñ ÐºÐ¸, ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ
Ð½ÐµÑ Ð´Ð¾Ð±Ð½Ð¾ Ð¾Ð¿Ð¸Ñ Ñ Ð²Ð°Ñ Ñ Ñ ÐµÑ ÐµÐ· Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ñ
def. Ð Ð±Ñ Ð°Ñ ÐµÐ½Ð¸Ðµ Ðº Ð±Ð¸Ð±Ð»Ð¸Ð¾ÐµÐºÐ°Ð¼ Ð¿Ñ Ð¾Ð¸Ñ Ñ Ð¾Ð´Ñ
Ð¿Ð¾ Ð¸Ñ Ð¸Ð¼ÐµÐ½Ð°Ð¼.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

list_names ::= name list_names

name ::= word | string

url â Ñ Ð¿Ð¸Ñ Ð¾Ð URL;
morphology â Ñ Ð¿Ð¸Ñ Ð¾Ð Ð¼Ð¾Ñ Ñ Ð¾Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¸Ñ Ñ Ð»Ð¾Ð²Ð°Ñ ÐµÐ¹;
category â Ð³Ñ Ñ Ð¿Ð¿Ð° ÐºÐ°Ñ ÐµÐ³Ð¾Ñ Ð¸Ð¹;
useragent â Ñ Ð¿Ð¸Ñ Ð¾Ð Ñ Ð·ÐµÑ Ð°Ð³ÐµÐ½Ñ Ð¾Ð²;
mime â Ñ Ð¿Ð¸Ñ Ð¾Ð Ñ Ð¸Ð¿Ð¾Ð² ÐºÐ¾Ð½Ñ ÐµÐ½Ñ Ð°;
network â Ñ Ð¿Ð¸Ñ Ð¾Ð Ñ ÐµÑ ÐµÐ¹/IP-Ð°Ð´Ñ ÐµÑ Ð¾Ð²;
time â Ð±Ð¸Ð±Ð»Ð¸Ð¾ÑÐµÐºÐ° Ñ Ð¿Ñ Ð¾Ð¼Ð¾ÐµÐ¶Ñ Ñ ÐºÐ°Ð¼Ð¸
Ð²Ñ ÐµÐ¼ÐµÐ½Ð¸.

Ð Ñ Ð¸Ð¼ÐµÑ :

```
DENY src.ip = lib.network("Bad ips", "Test ips")
DENY dst.ip = lib.network("Bad ips")
DENY dst.ip = lib.url("Bad urls") % Ð² Ð´Ð°Ð½Ð½Ð¾Ð¼ Ñ Ð»Ñ Ð°Ðµ Ð´Ð¾Ð¼ÐµÐ½Ñ Ð±Ñ Ð´Ñ Ñ Ñ ÐµÐ·Ð¾Ð

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)
```

Ð Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ñ

Version: 7.x
Product: NGFW
Ð Ñ Ð°Ð¹Ð»Ð°Ñ Ð¿Ð¾Ð¸Ñ Ð¸Ðº Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ñ (def) Ñ Ð»Ñ Ð¶Ð°Ñ Ð´Ð»Ñ Ð¾Ð±Ñ ÐµÐ´Ð¸Ð½ÐµÐ½Ð¸Ñ
Ð½Ð°Ð±Ð¾Ñ Ð¾Ð² Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ð¸Ð»Ð¸ Ð´ÐµÐ¹Ñ Ð²Ð¸Ð¹.
Ð Ð°Ð¶Ð´Ð¾Ðµ Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ðµ Ð´Ð¾Ð»Ð¶Ð½Ð¾ Ð¸Ð¼ÐµÑ Ñ Ñ Ð½Ð°Ñ Ð¸Ð¼Ñ, Ð¿Ð¾ÑÐ»Ðµ
Ð½Ð¸Ð°ÐºÐ°Ð»Ñ Ð½Ðµ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ð¸Ð¼Ñ Ð¸Ð¸ Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ñ Ð¿Ð¾Ð¸Ñ Ð¾Ð¸Ð¸
Ð¿Ð¾Ñ Ð¾Ð¾Ð¸Ñ Ð¾Ð¸Ñ Ð¾Ð¸Ñ Ð¾ Ð½Ð¸Ñ Ð¾Ð¸Ñ Ð¸Ð¾Ð¸Ð¾Ð¸Ñ Ð¾Ð±Ñ Ð¸Ð¾Ñ Ð¸Ð¸Ñ Ð¸Ð¸Ñ Ð¸Ð¸

condition ::= name '=' value

label_name ::= atom

atom ::= [a-z][0-9a-zA-Z_]+

def scenario_cond

ÐiÐ¿Ð¸Ñ Ð¾Ðº Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ . Ð Ð°Ð¶Ð´Ð¾Ðµ Ñ Ñ Ð»Ð¾Ð²Ð¸Ðµ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ Ð¾Ð±Ñ Ñ Ð½Ð¾ Ð¿Ð¸Ñ ÐµÑ Ñ Ñ Ð² Ð¾Ð´Ð½Ð¾Ð¹ Ñ Ñ Ñ Ð¾ÐºÐµ, Ð½Ð¾ Ð¿Ñ Ð¸ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾Ñ Ñ Ð¸ Ñ Ñ Ð»Ð¾Ð²Ð¸Ðµ Ð¼Ð¾Ð¶ÐµÑ Ð±Ñ Ñ Ñ Ñ Ð°Ð·Ð±Ð¸Ñ Ð¾ Ð½Ð° Ñ Ñ Ñ Ð¾ÐºÐ¸ Ñ Ð¿Ð¾Ð¼Ð¾Ñ Ñ Ñ Ñ Ð¿ÐµÑ Ð¸Ð°Ð»Ñ Ð½Ð¾Ð³Ð¾ Ñ Ð¸Ð¼Ð²Ð¾Ð»Ð° â Ð¾Ð±Ñ Ð°Ñ Ð½Ð¾Ð³Ð¾ Ñ Ð»ÐµÑ Ð° ("\").

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ðµ Ð¿ÐµÑ ÐµÐ¼ÐµÐ½Ð½Ñ Ñ . ÐiÐ»Ñ Ð¶Ð¸Ñ Ð´Ð»Ñ Ð¿Ð¾Ð´Ñ Ñ ÐµÑ Ð° Ð½ÐµÐºÐ¾Ñ Ð¾Ñ Ñ Ñ Ñ Ð¾Ð±Ñ Ñ Ð¸Ð¹ Ð·Ð° Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð½Ñ Ð¹ Ð¸Ð½Ñ ÐµÑ Ð²Ð°Ð» Ð²Ñ ÐµÐ¼ÐµÐ½Ð¸. Ð Ð»Ñ Ð¸Ð·Ð¼ÐµÐ½ÐµÐ½Ð¸Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ð¿Ñ ÐµÐ´Ð½Ð°Ð·Ð½Ð°Ñ ÐµÐ½Ñ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ *inc* Ð¸ *dec*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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 â Ñ Ñ Ð¾ Ð½Ð°Ñ Ð°Ð»Ñ Ð½Ð¾Ðµ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ Ð¿ÐµÑ ÐµÐ¼ÐµÐ½Ð½Ð¾Ð¹, Ðº ÐºÐ¾Ñ Ð¾Ñ Ð¾Ð¼Ñ Ð²Ð° Ð²ÐµÑ Ð½ÐµÑ Ñ Ñ Ð¿Ð¾ Ð¸Ñ Ñ ÐµÑ ÐµÐ½Ð¸Ð¸ Ð²Ñ ÐµÐ¼Ñ *window*;

key â Ð¿Ð¾Ð»Ðµ Ð¸Ð»Ð¸ Ñ Ð¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð»ÐµÐ¹, Ð¿Ð¾ ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¼ Ð³Ñ Ñ Ð¿Ð¿Ð¸Ñ Ñ Ñ Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ Ð¿ÐµÑ Ð¸ÐµÐ¼ÐµÐ½Ð½Ð¾Ð¹ (Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ).

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

Version: 7.x
Product: NGFW
ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° (properties) â Ñ Ñ Ð¾ Ð½ÐµÐºÐ¾Ñ Ð°Ñ Ñ Ñ Ð±Ñ Ñ Ñ Ð¿Ñ Ð°Ð²Ð°, Ð½Ð°Ð¿Ñ Ñ Ð¼Ð¾Ñ *name* Ñ Ð»Ñ *enabled*. Ð Ð½Ð¸ Ñ Ñ Ð¿Ð¾Ð»Ñ Ð·Ñ Ñ Ñ Ñ Ñ Ñ Ð´Ð»Ñ Ð¿Ñ ÐµÐ´Ð¾Ñ Ñ Ð°Ð²Ð»ÐµÐ½Ð¸Ñ Ð´Ð¾Ð¿Ð¾Ð»Ð½Ð¸Ñ ÐµÐ»Ñ Ð½Ñ Ñ Ð²Ð¾Ð¹ÑÑ Ñ Ñ Ñ Ð¾Ñ Ð¾Ð¼ Ð²Ð¾Ð±Ñ Ð°Ð±Ð¾Ñ Ñ Ð¿Ð°Ð²Ð¸Ð».
ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ Ñ Ð²Ð¾Ð¹Ñ Ñ Ð² Ñ Ð¾Ñ Ð½Ð¾Ð¾ Ñ Ð°ÐºÐ¾Ð¹ Ð¶Ðµ, ÐºÐ°Ðº Ñ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ð¹.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

property = prop_name | prop_name '(' list_params ')'

prop_name ::= name | desc | id | rule_log | enabled | scenario

list_params ::= value ',' list_params

name Ð¸ desc

Ð Ñ Ñ Ð¸Ð±Ñ Ñ Ñ *Ð¸Ñ Ð¸ Ð¾Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ* Ð´Ð»Ñ Ð¿Ð¾Ð»Ñ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

Name ::= name '(' string|word ')'

Description ::= desc '(' string ')'

Ð Ñ Ð¸Ñ ÐµÐ¼Ñ :

```
DENY hour = 9..18 category = News name("Ð Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ  News")  desc("Ð Ð°Ð¿Ñ ÐµÑ Ð¸Ñ Ñ  ÐºÐ°Ñ
```

enabled

Ð Ñ Ñ Ð¸Ð±Ñ Ñ , ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð²ÐºÐ»Ñ Ñ Ð°ÐµÑ Ð¸Ð»Ð¸ Ð²Ñ ÐºÐ»Ñ Ñ Ð°ÐµÑ Ñ Ð°Ð±Ð¾Ñ Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

Enable ::= enabled '(' boolean ')'

boolean ::= yes | no | true | false % (Ð¿Ð¾ Ñ Ð¼Ð¾Ð»Ñ Ð°Ð½Ð¸Ñ false)

rule_log

Ð£Ñ Ñ Ð°Ð½Ð°Ð²Ð»Ð¸Ð²Ð°ÐµÑ Ð°Ñ Ñ Ð¸Ð±Ñ Ñ Ð¶Ñ Ñ Ð½Ð°Ð»Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°. Ð Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ *session* Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°, Ð·Ð°Ñ Ð¸Ñ Ñ Ð¾Ñ dos-Ð°Ñ Ð°Ðº Ð¸ Ð¿Ñ Ð¾Ð¿Ñ Ñ ÐºÐ½Ð¾Ð¹ Ñ Ð¿Ð¾Ñ Ð¾Ð±Ð½Ð¾Ñ Ñ Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

Logging ::= rule_log '(' boolean | session ')'

LoggingFwRule ::= rule_log '(' boolean , interval, burst')'

boolean ::= yes | no | true | false % (Ð¿Ð¾ Ñ Ð¼Ð¾Ð»Ñ Ð°Ð½Ð¸Ñ no)

interval ::= "integer/[s,m,h,d]"

burst ::= integer

interval â Ñ Ñ ÐµÐ´Ð½ÐµÐµ Ñ Ð¸Ñ Ð»Ð¾ Ð¿Ð°ÐºÐµÑ Ð¾Ð², Ð¿Ð¾Ð¿Ð°Ð´Ð°Ñ Ñ Ð¸Ñ Ð¿Ð¾Ð´ Ñ Ñ Ð»Ð¾Ð²Ð¸Ðµ *limit* Ð² ÐµÐ´Ð¸Ð½Ð¸Ñ Ñ Ð²Ñ ÐµÐ¼ÐµÐ½Ð¸ (1/s, 1/m, 1/h, 1/d) , default = 3/h;
burst â Ð¼Ð°ÐºÑ Ð¸Ð¼Ð°Ð»Ñ Ð½Ð¾Ðµ Ñ Ð¸Ñ Ð»Ð¾ Ð¿Ð°ÐºÐµÑ Ð¾Ð², Ð¿Ð¾Ð¿Ð°Ð´Ð°Ñ Ñ Ð¸Ñ Ð² Ð¿Ð¾Ð´ Ñ Ñ Ð»Ð¾Ð²Ð¸Ðµ *limit* Ð·Ð° Ð¾Ð´Ð¸Ð½ Ñ Ð°Ð· (default = 5).

profile

Ð£Ñ Ñ Ð°Ð½Ð°Ð²Ð»Ð¸Ð²Ð°ÐµÑ Ð¿Ñ Ð¾Ñ Ð¸Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

certificate

ÐiÐµÑ Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ , Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ñ ÐµÐ¼Ñ Ð¹ Ð´Ð»Ñ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶ÐºÐ¸ HTTPS-Ñ Ð¾ÐµÐ´Ð¸Ð½ÐµÐ½Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Certificate ::= certificate '(' certificate_name ')'

certificate_name ::= string | word

gateway

Ð¨Ð»Ñ Ð·. Ð Ð¼Ñ Ð¾Ð´Ð½Ð¾Ð³Ð¾ Ð¸Ð· Ñ Ñ Ñ ÐµÑ Ñ Ð²Ñ Ñ Ñ Ð¸Ñ Ñ Ð»Ñ Ð·Ð¾Ð². Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸, Ð¸ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ ".

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°

reject_with

Ð£Ñ Ñ Ð°Ð½Ð°Ð²Ð»Ð¸Ð²Ð°ÐµÑ Ñ Ð¿Ð¾Ñ Ð¾Ð±, Ñ Ð¿Ð¾Ð¼Ð¾Ñ Ñ Ñ ÐºÐ¾Ñ Ð¾Ñ Ð¾Ð³Ð¾ Ð±Ñ Ð´ÐµÑ Ð±Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²Ð°Ñ Ñ Ñ Ñ Ñ Ñ Ð°Ñ Ð¸Ðº. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

fragmented

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð½Ð° Ñ Ñ Ð°Ð³Ð¼ÐµÐ½Ñ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ð¾Ñ Ñ Ñ Ð¿Ð°ÐºÐµÑ Ð¾Ð². Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

yes â Ð¿Ñ Ð¾Ð²ÐµÑ Ñ Ñ Ñ Ñ Ñ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ñ Ñ Ð°Ð³Ð¼ÐµÐ½Ñ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ Ðµ Ð¿Ð°ÐºÐµÑ Ñ ;
no â Ð¿Ñ Ð¾Ð²ÐµÑ Ñ Ñ Ñ Ñ Ñ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð½ÐµÑ Ñ Ð°Ð³Ð¼ÐµÐ½Ñ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ Ðµ Ð¿Ð°ÐºÐµÑ Ñ ;
â , ÐµÑ Ð»Ð¸ Ñ Ð²Ð¾Ð¹Ñ Ñ Ð²Ð¾ *fragmented* Ð½Ðµ Ñ ÐºÐ°Ð·Ð°Ð½Ð¾, Ñ Ð¾ Ð±Ñ Ð´Ñ Ñ Ð¿Ñ Ð¾Ð²ÐµÑ Ñ Ñ Ñ Ñ Ñ Ð²Ñ Ðµ Ð¿Ð°ÐºÐµÑ Ñ .

ips_profile

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

Ð£Ñ Ñ Ð°Ð½Ð°Ð²Ð»Ð¸Ð²Ð°ÐµÑ Ð¿Ñ Ð¾Ñ Ð¸Ð»Ñ IPS. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

l7_profile

Ð£Ñ Ñ Ð°Ð½Ð°Ð²Ð»Ð¸Ð²Ð°ÐµÑ Ð¿Ñ Ð¾Ñ Ð¸Ð»Ñ Ð¿Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¼ÐµÐ¶Ñ ÐµÑ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

L7_Profile ::= l7_profile '(' string | word ')'

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL

block_invalid_cert

Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ðµ Ñ Ð°Ð¹Ñ Ð¾Ð² Ñ Ð½ÐµÐºÐ¾Ñ Ñ ÐµÐºÑ Ð½Ñ Ð¼Ð¸Ñ ÐµÑ Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð°Ð¼Ð¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

check_revoc_cert

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð¿Ð¾ Ñ Ð¿Ð¸Ñ ÐºÑ Ð¾Ñ Ð¾Ð·Ð²Ð°Ð½Ð½Ñ Ñ Ñ ÐµÑ Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¾Ð². Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

block_expired_cert

Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ñ ÐµÑ Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¾Ð² Ñ Ð¸Ñ Ñ ÐµÐºÑ Ð¸Ð¼ Ñ Ñ Ð¾ÐºÐ¾Ð¼ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

block_self_signed_cert

Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ñ²ÐºÐ° Ñ Ð°Ð¼Ð¾Ð¿Ð¾Ð´Ð¿Ð¸ÑÐ°Ð½Ð½Ñ Ñ Ñ ÐµÑ Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¾Ñ². Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ SSL.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

ssl_profile

Ð Ñ Ð¾Ñ Ð¸Ð»Ñ SSL. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ SSL, reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸, Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

ssl_forward_profile

Ð Ñ Ð¾Ñ Ð¸Ð»Ñ Ð¿ÐµÑ ÐµÑ Ñ Ð»ÐºÐ¸ SSL. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð½Ñ SSL.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

profile_id ::= integer

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð» Ð¿Ñ Ð¾ÐºÑ Ñ ÐºÐ½Ð¾Ð¹ Ñ Ð¿Ð¾Ñ Ð¾Ð±Ð½Ð¾Ñ Ñ Ð¸

bandwidth_pool

Ð Ð¾Ð»Ð¾Ñ Ð° Ð¿Ñ Ð¾Ð¿Ñ Ñ ÐºÐ°Ð½Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð±Ð½Ð¾Ñ Ñ Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

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

Ð Ñ Ð¸Ð¼ÐµÑ :

```
scenario = "Ðiѕ ÐµÐ½Ð°Ñ Ð¸Ð³Ð±Ð½Ð°Ñ Ñ Ð¶ÐµÐ½Ð¸Ñ  torrent" bandwidth_pool("1 Mbps") enable(true)
```

Ðiвð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð·Ð°Ñ Ð¸Ñ Ñ Ð¿ð¾Ñ Ñ Ð³Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°

mark

Ð Ð°Ñ ÐºÐ¸Ñ Ð¾Ð²ÐºÐ°. Ð¢ÐµÐºÑ Ñ Ñ ÐµÐ³Ð°, ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð¼Ð°Ñ ÐºÐ¸Ñ Ñ ÐµÑ Ð¿Ð¸Ñ Ñ Ð¼Ð¾. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð·Ð°Ñ Ð¸Ñ Ñ Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

mark_hdr

Ð Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº. Ð Ð¾Ð»Ðµ, ÐºÑ Ð´Ð° Ð¿Ð¾Ð¼Ð¼ÐµÑ Ð°ÐµÑ Ñ Ñ Ñ ÐµÐ³ Ð¼Ð°Ñ ÐºÐ¸Ñ Ð¾Ð²ÐºÐ¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð·Ð°Ñ Ð¸Ñ Ñ Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

antispam_kav

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð°Ð½Ñ Ð¿Ð°Ð¼Ð¼Ð¾Ð². Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð·Ð°Ñ Ð¸Ñ Ñ Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

antispam_usergate

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð°Ð½Ñ Ð¿Ð°Ð¼Ð¼Ð¾Ð¼ UserGate.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð·Ð°Ñ Ð¸Ñ Ñ Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ð°Ð»Ð¸ Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð²Ð°Ð½Ð¸Ñ SSL block_invalid_cert

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

boolean ::= yes | no | true | false

Ð Ñ Ð¸Ð¼ÐµÑ :

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

dnsbl

DNSBL-Ð¿Ñ Ð¾Ð²ÐµÑ ÐºÐ° (Ñ Ð¾Ð»Ñ ÐºÐ¾ SMTP). Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð·Ð°Ñ Ð¸Ñ Ñ Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

Ð Ñ Ð¸Ð¼ÐµÑ :

```
DENY service = SMTP dnsbl(yes)
```

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸

target_ip

Ð Ð´Ñ ÐµÑ Ð½Ð°Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ DNAT, ÐµÑ Ð»Ð¸ Ð·Ð°Ð´Ð°Ð½Ð¾ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ðµ *dnat* Ð¸Ð»Ð¸ *port_mapping*.

Ð Ð¾Ð²Ð°Ñ IP-Ñ ÐµÑ Ñ /Ð¼Ð°Ñ ÐºÐ°, ÐµÑ Ð»Ð¸ Ð·Ð°Ð´Ð°Ð½Ð¾ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸Ðµ *netmap.*

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð Ñ Ð¸Ð¼ÐµÑ :

```
PASS target_ip("192.168.1.20") dnat
```

target_snat

Ð ÐºÐ»Ñ Ñ ÐµÐ½Ð¸Ðµ SNAT. Ð Ñ Ð¸ Ð²ÐºÐ»Ñ Ñ ÐµÐ½Ð¸Ð¸ Ð´Ð°Ð½Ð½Ð¾Ð¹ Ð¾Ð¿Ñ Ð¸Ð¸ UserGate Ð±Ñ Ð´ÐµÑ Ð¸Ð·Ð¼ÐµÐ½Ñ Ñ Ñ Ð°Ð´Ñ ÐµÑ Ð¸Ñ Ñ Ð¾Ñ Ð½Ð¸ÐºÐ° Ð² Ð¿Ð°ÐºÐµÑ Ð°Ñ Ð¸Ð· Ð²Ð½ÐµÑ Ð½ÐµÐ¹ Ñ ÐµÑ Ð¸ Ð½Ð° Ñ Ð²Ð¾Ð¹ IP-Ð°Ð´Ñ ÐµÑ .

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

snat_target_ip

SNAT IP (Ð²Ð½ÐµÑ Ð½Ð¸Ð¹ Ð°Ð´Ñ ÐµÑ). Ð¯Ð²Ð½Ð¾ Ñ ÐºÐ°Ð·Ñ Ð²Ð°ÐµÑ IP-Ð°Ð´Ñ ÐµÑ , Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð±Ñ Ð´ÐµÑ Ð·Ð°Ð¼ÐµÐ½ÐµÐ½ Ð°Ð´Ñ ÐµÑ Ð¸Ñ Ñ Ð¾Ñ Ð½Ð¸ÐºÐ° Ð¿Ñ Ð¸ Ð¿Ð¾Ð´Ð¼ÐµÐ½Ðµ Ð°Ð´Ñ ÐµÑ Ð¾Ð² Ð¿Ð°ÐºÐµÑ Ð¾Ð².

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

SnatTargetIp ::= snat_target_ip '(' ip_address ')'

port_map

Ð Ð¾Ñ Ñ -Ñ Ð¾Ñ Ð²Ð°Ñ Ð´Ð¸Ð½Ð³. Ð ÐµÑ ÐµÐ¾Ð¿Ñ ÐµÐ´ÐµÐ»ÐµÐ½Ð¸Ðµ Ð¿Ð¾Ñ Ñ Ð¾Ð² Ð¿Ñ Ð±Ð»Ð¸ÐºÑ ÐµÐ¼Ñ Ñ Ñ ÐµÑ Ð²Ð¸Ñ Ð¾Ð².
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

protocol ::= tcp | udp | smtp | smpts

port_from ::= integer

port_to ::= integer

port_from â Ð½Ð¾Ð¼ÐµÑ TCP/UDP-Ð¿Ð¾Ñ Ñ Ð°, Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ð¸ Ñ Ð»Ñ Ñ Ð·Ð°Ð¿Ñ Ð¾Ñ Ñ ;
port_to â Ð½Ð¾Ð¼ÐµÑ TCP/UDP-Ð¿Ð¾Ñ Ñ Ð°, Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð±Ñ Ð´Ñ Ñ Ð¿ÐµÑ ÐµÑ Ñ Ð»Ð°Ñ Ñ Ñ Ñ Ð·Ð°Ð¿Ñ Ð¾Ñ Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»ÐµÐ¹ Ð½Ð° Ð²Ð½Ñ Ñ Ñ ÐµÐ½Ð½Ð¸Ð¹ Ð¿Ñ Ð±Ð»Ð¸ÐºÑ ÐµÐ¼Ñ Ð¹ Ñ ÐµÑ Ð²ÐµÑ .

direction

Ð Ð°Ð¿Ñ Ð°Ð²Ð»ÐµÐ½Ð¸Ðµ Ð¿Ð¾Ð´Ð¼ÐµÐ½Ñ Ñ ÐµÑ ÐµÐ¹.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

input â Ð²Ñ Ð¾Ð´Ñ Ñ Ð¸Ð¹, Ð¿Ð¾Ð´Ð¼ÐµÐ½Ñ ÐµÑ Ñ IP-Ñ ÐµÑ Ñ Ð½Ð°Ð·Ð½Ð°Ñ . Ð Ñ Ð´Ñ Ñ Ð¸Ð·Ð¼ÐµÐ½ÐµÐ½Ñ IP-Ð°Ð´Ñ ÐµÑ Ð° Ð½Ð°Ð·Ð½Ð°Ñ Ð² Ñ Ñ Ð°Ñ Ð¸ÐºÐµ;
ouput â Ð¸Ñ Ñ Ð¾Ð´Ñ Ñ Ð¸Ð¹, Ð¿Ð¾Ð´Ð¼ÐµÐ½Ñ ÐµÑ Ñ IP-Ñ ÐµÑ Ñ Ð¸Ñ Ñ Ð¾Ð½Ð°Ð°. Ð Ñ Ð´Ñ Ñ Ð¸Ð·Ð¼ÐµÐ½ÐµÐ½Ñ IP-Ð°Ð´Ñ ÐµÑ Ð° Ð¸Ñ Ñ Ð¾Ð½Ð°Ð° Ð² Ñ Ñ Ð°Ñ Ð¸ÐºÐµ.

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» DNS-Ð¿Ñ Ð¾ÐºÑ Ð¸

dns_server

ÐiÐ¿Ð¸Ñ Ð¾Ðº IP-Ð°Ð´Ñ ÐµÑ Ð¾Ð² DNS-Ñ ÐµÑ Ð²ÐµÑ Ð¾Ð².
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» DNS-Ð¿Ñ Ð¾ÐºÑ Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸

cert_auth_enabled

Ð Ñ Ñ ÐµÐ½Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¸Ñ Ð¿Ð¾ Ñ Ð¸Ð¾ Ñ ÐµÑ Ñ Ð¸ÐºÐ°Ñ Ñ .
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

is_https

Ð ÐºÐ»Ñ Ñ ÐµÐ½Ð¸Ðµ Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶ÐºÐ¸ HTTPS. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

rewrite_path

Ð Ð¾Ð´Ð¼ÐµÐ½Ð° Ð¿Ñ Ñ ÐµÐ¹. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL block_invalid_cert

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

path_from â Ð¸Ð·Ð¼ÐµÐ½Ð¸Ñ Ñ Ñ (Ð´Ð¾Ð¼ÐµÐ½ Ð¸/Ð¸Ð»Ð¸ Ð¿Ñ Ñ Ñ URL, ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ Ñ Ñ ÐµÐ±Ñ ÐµÑ Ñ Ñ Ð¸Ð·Ð¼ÐµÐ½Ð¸Ñ Ñ);
path_to â Ð¸Ð·Ð¼ÐµÐ½Ð¸Ñ Ñ Ð½Ð° (Ð´Ð¾Ð¼ÐµÐ½ Ð¸/Ð¸Ð»Ð¸ Ð¿Ñ Ñ Ñ URL, Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ Ñ Ñ ÐµÐ±Ñ ÐµÑ Ñ Ñ Ð·Ð°Ð¼ÐµÐ½Ð¸Ñ Ñ Ñ Ñ Ð°Ñ Ñ Ðµ).

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°

icon

Ð ÐºÐ¾Ð½ÐºÐ°, ÐºÐ¾Ñ Ð¾Ñ Ð°Ñ Ð±Ñ Ð´ÐµÑ Ð¾Ñ Ð¾Ð±Ñ Ð°Ð¶Ð°Ñ Ñ Ñ Ñ Ð½Ð° Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ðµ Ð´Ð»Ñ Ð´Ð°Ð½Ð½Ð¾Ð¹ Ð·Ð°ÐºÐ»Ð°Ð´ÐºÐ¸.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

additional_url

Ð Ñ Ð¿Ð¾Ð¼Ð¾Ð³Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ðµ URL, Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ñ Ðµ Ð´Ð»Ñ Ñ Ð°Ð±Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¾Ð²Ð½Ð¾Ð³Ð¾ URL, Ð½Ð¾ ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ Ð½ÐµÑ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾Ñ Ñ Ð¸ Ð¿Ñ Ð±Ð»Ð¸ÐºÐ¾Ð²Ð°Ñ Ñ Ð´Ð»Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»ÐµÐ¹.

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

rdp_check_session_alive

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ð°Ð²Ñ Ð¾Ñ Ð¸Ð·Ð°Ñ Ð¸Ð¸ Ð´Ð»Ñ RDP-Ñ ÐµÑ Ñ Ð¸Ð¹.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

transparent_auth

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL block_invalid_cert

Ð ÐºÐ»Ñ Ñ ÐµÐ½Ð¸Ðµ Ð¿Ñ Ð¾Ð·Ñ Ð°Ñ Ð½Ð¾Ð¹ Ð°Ñ Ñ ÐµÐ½Ñ Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¸Ð¸ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

TansparentAuth ::= transparent_auth '(' boolean ')'

boolean ::= yes | no | true | false

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð²ÐµÐ±-Ð±ÐµÐ·Ð¾Ð¿Ð°Ñ Ð½Ð¾Ñ Ñ Ð¸ (Safe browsing)

enable_adblock

Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ñ ÐµÐºÐ»Ð°Ð¼Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

url_list_exclusions

URL-Ñ Ð¿Ð¸Ñ Ð¾Ðº Ñ Ð°Ð¹Ñ Ð¾Ð² Ð¸Ñ ÐºÐ»Ñ Ñ ÐµÐ½Ð¸Ð¹, Ð´Ð»Ñ ÐºÐ¾Ñ Ð¾Ñ Ñ Ñ Ð±Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²Ð°Ñ Ñ Ñ ÐµÐºÐ»Ð°Ð¼Ñ Ð½Ðµ Ñ Ñ ÐµÐ±Ñ ÐµÑ Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

list_libs :: = lib_name ','' list_libs

lib_name ::= word | string

enable_injector

Ð Ð¾Ð·Ð²Ð¾Ð»Ñ ÐµÑ Ð²Ñ Ñ Ð°Ð²Ð¸Ñ Ñ Ð¿Ñ Ð¾Ð¸Ð·Ð²Ð¾Ð»Ñ Ð½Ñ Ð¹ ÐºÐ¾Ð´ Ð²Ð¾ Ð²Ñ Ðµ Ð²ÐµÐ±-Ñ Ñ Ñ Ð°Ð½Ð¸Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

custom_injector

Ð Ð¾Ð´ Ð¸Ð½Ð¶ÐµÐºÑ Ð¾Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

ÐiustomInjector ::= custom_injector '(' string ')'

safe_search

Ð¤Ñ Ð½ÐºÑ Ð¸Ñ Ð±ÐµÐ·Ð¾Ð¿Ð°Ñ Ð½Ð¾Ð³Ð¾ Ð¿Ð¾Ð¸Ñ ÐºÐ°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

search_history_logging

Ð Ð°Ð¿Ð¸Ñ Ñ Ð² Ð¶Ñ Ñ Ð½Ð°Ð» Ð¿Ð¾Ð¸Ñ ÐºÐ¾Ð²Ñ Ñ Ð·Ð°Ð¿Ñ Ð¾Ñ Ð¾Ð² Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»ÐµÐ¹.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

social_sites_block

Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²Ð°Ñ Ð¿Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ñ Ð² Ð¿Ð¾Ð¿ÑÐ»Ñ Ñ Ñ Ð¾Ñ Ñ Ð¸Ð°Ð»Ñ Ð½Ñ Ñ Ñ ÐµÑ Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð° BYOD

max_device_number

Ð Ð°ÐºÑ Ð¸Ð¼Ð°Ð»Ñ Ð½Ð¾Ð¼ ÐºÐ¾Ð»Ð¸Ñ ÐµÑ Ñ Ð²Ð¾ Ñ Ñ Ñ Ñ Ð¾Ð¹Ñ Ñ Ð², Ñ ÐºÐ¾Ñ Ð¾Ñ Ñ Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ð¼Ð¾Ð¶ÐµÑ Ñ Ð°Ñ Ñ Ð´Ð¾Ñ Ñ Ñ Ð¿ Ð² Ñ ÐµÑ Ñ .
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ Ð½Ð¾Ñ Ð½Ð¾Ð¼ Ð¾ÐºÐ¾Ð½ÑÐ°Ð½Ð¸Ñ BYOD.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

max_active_device_number

Ð Ð°ÐºÑ Ð¸Ð¼Ð°Ð»Ñ Ð½Ð¾Ð¼ÐµÑ ÐºÐ¾Ð»Ð¸Ñ Ñ ÐµÑ Ñ Ñ Ñ Ð¾Ð±Ñ Ñ Ð², Ñ ÐºÐ¾Ñ Ð¾Ñ Ñ Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ð¾Ð´Ð½Ð¾Ð²Ñ ÐµÐ¼ÐµÐ½Ð½Ð¾
Ð¼Ð¾Ð¶ÐµÑ Ð¿Ð¾Ð»Ñ Ñ Ð°Ñ Ñ Ð´Ð¾Ñ Ñ Ð¿Ð¾Ð»Ñ Ð² Ñ ÐµÑ .
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» BYOD.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

device_type

Ð¢Ð¸Ð¿ Ñ Ñ Ñ Ð¾Ð¹Ñ Ñ Ð², Ð´Ð»Ñ ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¿Ñ Ð¸Ð¼ÐµÐ½Ñ ÐµÑ Ñ
Ð´Ð°Ð½Ð½Ð¾Ð¼Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ñ Ð¿Ð¾Ð»Ñ Ð¸Ð¾Ð»Ñ ÐºÐ¾ BYOD.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» BYOD.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

approving_required

Ð Ð¾Ð´Ð¾Ð²ÐµÑ Ð¶Ð´ÐµÐ½Ð¸Ðµ Ð°Ð´Ð¼Ð¸Ð½Ð¸Ñ Ñ Ñ Ð°Ñ Ð¾Ñ Ð°.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» BYOD.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¸Ð½Ñ Ðµ¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSH

block_ssh_shell

Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ñ Ð´Ð°Ð»Ñ Ð½Ð½Ð¾Ð³Ð¾ Ð·Ð°Ð¿Ñ Ñ ÐºÐ° shell.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ
Ð¿Ñ Ð°Ð²Ð» Ð¸Ð½Ñ Ðµ¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSH.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

boolean ::= yes | no | true | false

block_ssh_exec

Đ Đ»Đ¾ĐºĐ¸Ñ Đ¾Đ²ĐºĐ° Ñ Đ´Đ°Đ»Ñ Đ½Đ½Đ¾Đ³Đ¾ Đ²Ñ Đ¿Đ¾Đ»Đ½ĐµĐ½Đ¸Ñ ĐºĐ¾Đ¼Đ°Đ½Đ´ Đ¿Đ¾ SSH. Đ ĐµĐ¹Ñ Ñ Đ²Đ¸Ñ ĐµĐ»Ñ Đ½Đ¾ Ñ Đ¾Đ»Ñ ĐºĐ¾ Đ´Đ»Ñ Đ¿Ñ Đ°Đ²Đ¸Đ» Đ¸Đ½Ñ Đ¿ĐµĐºÑ Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ SSH.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

boolean ::= yes | no | true | false

block_sftp

Đ Đ»Đ¾ĐºĐ¸Ñ Đ¾Đ²ĐºĐ° Ñ Đ¾ĐµĐ´Đ¸Đ½ĐµĐ½Đ¸Ñ SFTP (Secure File Transfer Protocol). Đ ĐµĐ¹Ñ Ñ Đ²Đ¸Ñ ĐµĐ»Ñ Đ½Đ¾ Ñ Đ¾Đ»Ñ ĐºĐ¾ Đ´Đ»Ñ Đ¿Ñ Đ°Đ²Đ¸Đ» Đ¸Đ½Ñ Đ¿ĐµĐºÑ Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ SSH.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

boolean ::= yes | no | true | false

ssh_command

Đ Đ¾Đ¼Đ°Đ½Đ´Đ° linux, ĐºĐ¾Ñ Đ¾Ñ Ñ Ñ Ñ Ñ ĐµĐ±Ñ ĐµÑ Ñ Ñ Đ¿ĐµÑ ĐµĐ´Đ°Ñ Ñ , Đ² Ñ Đ¾Ñ Đ¼Đ°Ñ Đµ ssh user@host 'command'.

Đ ĐµĐ¹Ñ Ñ Đ²Đ¸Ñ ĐµĐ»Ñ Đ½Đ¾ Ñ Đ¾Đ»Ñ ĐºĐ¾ Đ´Đ»Ñ Đ¿Ñ Đ°Đ²Đ¸Đ» Đ¸Đ½Ñ Đ¿ĐµĐºÑ Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ SSH.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

ĐiĐ²Đ¾Đ¹Ñ Ñ Đ²Đ° Đ¿Ñ Đ°Đ²Đ¸Đ» VPN-Ñ ĐµÑ Đ²ĐµÑ Đ°

auth_profile

Đ Ñ Đ¾Ñ Đ¸Đ»Ñ Đ°Ñ Ñ ĐµĐ½Ñ Đ¸Ñ Đ¸ĐºĐ°Ñ Đ¸Đ¸. Đ ĐµĐ¹Ñ Ñ Đ²Đ¸Ñ ĐµĐ»Ñ Đ½Đ¾Ñ Đ¾Đ»Ñ ĐºĐ¾ Đ´Đ»Ñ Đ¿Ñ Đ°Đ²Đ¸Đ» VPN-Ñ ĐµÑ Đ²ĐµÑ Đ°.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

vpn_network

ÐiÐµÑ Ñ VPN. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-Ñ ÐµÑ Ð²ÐµÑ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

interface

Ð Ð½Ñ ÐµÑ Ñ ÐµÐ¹Ñ VPN. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-Ñ ÐµÑ Ð²ÐµÑ Ð° Ð¸ VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°

server_address

IP-Ð°Ð´Ñ ÐµÑ Ñ ÐµÑ Ð²ÐµÑ Ð°. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

password

Ð Ð°Ñ Ð¾Ð»Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

last_error

Ð Ð¾Ñ Ð»ÐµÐ´Ð½Ñ Ñ Ð¾Ñ Ð¸Ð±ÐºÐ° VPN. Ð Ð½Ñ Ð¾Ñ Ð¼Ð°Ñ Ð¸Ð¾Ð½Ð½Ð¾Ð¹ Ð¿Ð¾Ð»Ñ, Ð½ÐµÐ´Ð¾Ñ Ñ Ñ Ð¿Ð½Ð¾Ð¹ Ð´Ð»Ñ Ñ ÐµÐ´ÐºÑ Ð¸Ñ Ð¾Ñ²Ð°Ð½Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

status

ÐiÑ Ð°Ñ Ñ Ñ .Ð Ð½Ñ Ð¾Ñ Ð¼Ð°Ñ Ð¸Ð¾Ð½Ð½Ð¾Ð¼ Ð¿Ð¾Ð»Ðµ,
Ð½ÐµÐ´Ð¾Ñ Ñ Ñ Ð¿Ð½Ð¾Ðµ Ð´Ð»Ñ Ñ ÐµÐ´Ð°ÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ .
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ
Ð¿Ñ Ð°Ð²Ð¸Ð» VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ñ Ñ ÐµÐ½Ð°Ñ Ð¸ÐµÐ²

trigger

Ð¢Ð¸Ð¿ Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°Ð½Ñ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ .
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»
Ñ Ñ ÐµÐ½Ð°Ñ Ð¸ÐµÐ².

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

trigger_type ::= all_users | one_user

duration

Ð Ñ Ð¾Ð´Ð¾Ð»Ð¶Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾Ñ Ñ Ñ .Ð Ñ ÐµÐ¼Ñ Ð² Ð¼Ð¸Ð½Ñ Ñ Ð°Ñ , Ð²
Ñ ÐµÑ ÐµÐ½Ð¸Ðµ ÐºÐ¾Ñ Ð¾Ñ Ð¾Ð³Ð¾ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ Ð±Ñ Ð´ÐµÑ
Ð°ÐºÑ Ð¸Ð²Ð½Ñ Ð¼ Ð¿Ð¾Ñ Ð»Ðµ ÐµÐ³Ð¾ Ð°ÐºÑ Ð¸Ð²Ð°Ñ Ð¸Ð¸.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»
Ñ Ñ ÐµÐ½Ð°Ñ Ð¸ÐµÐ².

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

operation_mode

ÐiÑ ÐµÐ½Ð°Ñ Ð¸Ð¹ Ñ Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°ÐµÑ Ð¿Ñ Ð¸Ð²Ð¿Ð°Ð´Ð°ÐµÐ¸Ð¸
Ð¾Ð´Ð½Ð¾Ð³Ð¾ Ð¸Ð»Ð¸ Ð²Ñ ÐµÑ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

mode ::= all | any

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ

scond_type

Ð¢Ð¸Ð¿ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

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

count_interval

Ð Ð¾Ð»Ð¸Ñ ÐµÑ Ñ Ð²Ð¾ Ñ Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°Ð½Ð¸Ð¹.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ð°Ñ ÐµÐ³Ð¾Ñ Ð¸Ñ URL" (url_category), "Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ðµ" (app), "Ð¢Ð¸Ð¿ ÐºÐ¾Ð½Ñ ÐµÐ½Ñ Ð°" (mime_type) Ð¸ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

max_event_count

Ð Ð½Ñ ÐµÑ Ð²Ð°Ð» Ð²Ñ ÐµÐ¼ÐµÐ½Ð¸ Ð² Ð¼Ð¸Ð½Ñ Ñ Ð°Ñ , Ð·Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð±Ñ Ð»Ð¾ Ð¾Ð±Ð½Ð°Ñ Ñ Ð¶ÐµÐ½Ð¾ *count_interval* Ñ Ñ Ð°Ð±Ð°Ñ Ñ Ð²Ð°Ð½Ð¸Ð¹.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ð°Ñ ÐµÐ³Ð¾Ñ Ð¸Ñ URL" (url_category), "Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ðµ" (app), "Ð¢Ð¸Ð¿ ÐºÐ¾Ð½Ñ ÐµÐ½Ñ Ð°" (mime_type) Ð¸ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

packet_size

Ð Ð°Ð·Ð¼ÐµÑ Ð¿Ð°ÐºÐµÑ Ð°. Ð Ð°Ð·Ð¼ÐµÑ Ð¿Ð°ÐºÐµÑ Ð° Ð² Ñ Ñ Ð°Ñ Ð¸ÐºÐµ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ð¿Ñ ÐµÐ²Ñ Ñ Ð¸Ð» Ñ ÐºÐ°Ð·Ð°Ð½Ð½Ð¾Ðµ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ð°Ð·Ð¼ÐµÑ Ð¿Ð°ÐºÐµÑ Ð°" (net_packet_size).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

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

64 â Ñ Ð°Ð·Ð¼ÐµÑ Ð² Ð±Ð°Ð¹Ñ Ð°Ñ ;
2MB â Ñ Ð°Ð·Ð¼ÐµÑ Ð² Ð¼ÐµÐ³Ð°Ð±Ð°Ð¹Ñ Ð°Ñ .

traffic_limit

Ð Ð±Ñ ÐµÐ¼ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ð³Ñ Ð°Ð½Ð¸Ñ ÐµÐ½Ð¸Ðµ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°" (traffic).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

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

period

Ð ÐµÑ Ð¸Ð¾Ð´ Ð²Ñ ÐµÐ¼ÐµÐ½Ð¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ð³Ñ Ð°Ð½Ð¸Ñ ÐµÐ½Ð¸Ðµ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°" (traffic).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

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

ips_tl

Ð£Ñ Ð¾Ð²ÐµÐ½Ñ Ñ Ð³Ñ Ð¾Ð·Ñ ÐiÐ Ð . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "ÐiÐ Ð " (ips).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð ÐµÑ Ð¾Ð´ Ð¿Ñ Ð¾Ð²ÐµÑ ÐºÐ¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ°
Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

method ::= ping | dns | get

health_result

Ð ÐµÐ·Ñ Ð»Ñ Ñ Ð°Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

result ::= positive | negative

health_request_timeout

Ð¢Ð°Ð¹Ð¼-Ð°Ñ Ñ Ð¿Ð¾Ð´ÐºÐ»Ñ Ñ ÐµÐ½Ð¸Ñ (Ñ ÐµÐº).
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

health_answer_timeout

Ð¢Ð°Ð¹Ð¼-Ð°Ñ Ñ Ð¾Ñ Ð²ÐµÑ Ð° (Ñ ÐµÐº). Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check), "Ð ÐµÑ Ð¾Ð´ Ð¿Ð¾Ð²ÐµÑ ÐºÐ¸ HTTP GET" (get).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

health_type_request

Ð¢Ð¸Ð¿ DNS-Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð»Ð¾Ð²Ð¸Ð¹ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ "Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸Ñ " (health_check), "Ð¼ÐµÑ Ð¾Ð´ Ð¿Ð¾Ð²ÐµÑ ÐºÐ¸ DNS" (dns).

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

type_name ::= word | string

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð° Ð¿Ñ Ð°Ð²Ð¸Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸

scheduler

Ð ÐµÑ Ð¾Ð´ Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ñ Ñ Ð¿Ð¸Ñ Ð¾Ðº Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

Scheduler ::= scheduler '(' balancing_type ')'

balancing_type ::= wrr | rr | lc | wlc

wrr â Weighted round robin;

rr â Round robin;

lc â Least connections;

wlc â Weighted least connections.

real_server

Ð ÐµÐ°Ð»Ñ Ð½Ñ Ðµ Ñ ÐµÑ Ð²ÐµÑ Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

mode ::= gate | masq | masq_snat

port ::= integer

weight ::= integer

gate â Ñ Ð»Ñ Ð·;

masq â Ð¼Ð°Ñ ÐºÐ°Ñ Ð°Ð´Ð¸Ð½Ð³;

masq_snat â Ð¼Ð°Ñ ÐºÐ°Ñ Ð°Ð´Ð¸Ð½Ð³ Ñ Ð¿Ð¾Ð´Ð¼ÐµÐ½Ð¾Ð¹ IP-Ð°Ð´Ñ ÐµÑ Ð° Ð¸Ñ Ñ Ð¾Ñ Ð½Ð¸ÐºÐ° (SNAT).

Ð Ñ Ð¸Ð¼ÐµÑ :

```
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

Ð Ð²Ð°Ñ Ð¸Ð½Ð¹ Ñ ÐµÐ¶Ð¸Ð¼ Ñ ÐµÑ Ð²ÐµÑ Ð°. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸

Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

mode ::= gate | masq | masq_snat

port ::= integer

monitor_kind

Ð ÐµÐ¶Ð¸Ð¼. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

kind ::= ping | connect | negotiate

monitor_service

ÐiÐµÑ Ð²Ð¸Ñ .Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

service ::= http | dns

monitor_request

Ð Ð°Ð¿Ñ Ð¾Ñ .Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

monitor_response

Ð Ð¶Ð¸Ð´Ð°ÐµÐ¼Ñ Ð¹ Ð¾Ñ Ð²ÐµÑ .Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

monitor_interval

Ð Ð½Ñ ÐµÑ Ð²Ð°Ð» Ð¿Ñ Ð¾Ð²ÐµÑ ÐºÐ¸. Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾
Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸
Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

monitor_timeout

Ð Ñ ÐµÐ¼Ñ Ð¾Ð¶Ð¸Ð´Ð°Ð½Ð¸Ñ . Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾
Ð´Ð»Ñ Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

monitor_failurecount

Ð§Ð¸Ñ Ð»Ð¾ Ð½ÐµÑ Ð´Ð°Ñ Ð½Ñ Ñ Ð¿Ð¾Ð¿Ñ Ñ Ð¾Ðº.
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ ÐµÐ»Ñ Ð½Ð¾ Ñ Ð¾Ð»Ñ ÐºÐ¾ Ð´Ð»Ñ
Ð¿Ñ Ð°Ð²Ð» Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ

Version: 7.x
Product: NGFW
Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ (action) â Ñ Ñ Ð¾Ñ Ð¾, Ñ Ñ Ð¾Ñ Ð±Ñ Ð´ÐµÑ
Ð²Ñ Ð¿Ð¾Ð»Ð½ÐµÐ½Ð½Ð¾Ñ,
ÐµÑ Ð»Ð¸ Ñ Ñ Ð»Ð¾Ð²Ð¸Ñ Ð² Ð¿Ñ Ð°Ð²Ð»Ðµ Ð¸Ñ Ñ Ð¸Ð½Ð½Ð¾ .Ð
ÐºÐ°Ñ ÐµÑ Ñ Ð²Ðµ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¾Ð² Ð¼Ð¾Ð³Ñ Ð¾Ñ
Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ Ñ Ñ Ñ ÐºÐ¾Ð½Ñ Ñ Ð°Ð½Ñ Ð½Ñ Ðµ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ ,
Ð¸Ð»Ð¸ Ð´Ð¸Ð½Ð°Ð¼Ð¸Ñ ÐµÑ ÐºÐ¸Ðµ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ñ Ð°Ð¼, Ð³Ð´Ðµ Ñ Ñ Ð¾
Ð¿Ñ ÐµÐ´Ñ Ñ Ð¼Ð¾Ñ Ñ ÐµÐ½Ð¸Ñ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ , ÐºÐ¾Ñ Ð¾Ñ Ñ Ðµ Ñ Ð°Ð·Ñ ÐµÑ Ð°Ñ Ñ Ñ Ñ Ð±ÐµÐ·
Ð²Ñ Ñ Ð¾Ð´Ð° Ð¸Ð· Ð¾Ð±Ñ Ð°Ð±Ð¾Ñ ÐºÐ¸ Ð¿Ð°Ð²Ð» Ð½Ð°
Ñ ÐµÐºÑ Ñ ÐµÐ¼ Ñ Ð»Ð¾Ð¼. *warning* Ñ Ñ Ð°Ð²Ð¸Ñ Ð¾Ñ Ð¼ÐµÑ ÐºÑ Ð¾ Ñ Ð¾Ð¼,
Ñ Ñ Ð¾ Ð½ÐµÐ¾Ð±Ñ Ð¾Ð´Ð¸Ð¼Ð¾ Ð²Ñ Ñ Ð°Ð²Ð¸Ñ Ñ ÐºÐ¾Ð´ Ñ
Ð¿Ñ ÐµÐ´Ñ Ð¿Ñ ÐµÐ¶Ð´ÐµÐ½Ð¸ÐµÐ¼ Ð² Ñ ÐµÐ»Ð¾ Ð¾Ñ Ð²ÐµÑ Ð°.

Ð Ñ Ð¸Ð¼ÐµÑ :

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

log_message

Ð Ð°Ð¿Ð¸Ñ Ð°Ñ Ñ Ð¾Ð¾Ð±Ñ ÐµÐ½Ð¸Ñ Ð² Ð¶Ñ Ñ Ð½Ð°Ð».

Ð Ñ Ð¸Ð¼ÐµÑ :

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

append

Ð Ð¾Ð±Ð°Ð²Ð¸Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Ðº HTTP-Ð·Ð°Ð¿Ñ Ð¾Ñ Ñ /Ð¾Ñ Ð²ÐµÑ Ñ .
Ði Ð¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´Ð¸ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÑ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾ÐºÐ¾Ð²
Ð¿Ñ Ð¸Ð²ÐµÐ´Ð½ Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶Ð¸Ð¸.

Ð ÐµÑ Ð²Ñ Ð¹ Ð¿Ð°Ñ Ñ Ð¼Ð¾Ð¶ÐµÑ Ñ Ð¾Ð¾Ð±Ñ, ÐµÑ Ð»Ð¸ Ð·Ð°Ð³Ð¾Ð»Ð¾Ðº Ð¾Ñ Ð¾Ñ Ñ Ð¾ Ð³Ð¾Ð»Ð¾Ð¹ Ð³Ñ Ñ Ð¿Ð¿Ðµ *request* Ð¸Ð»Ð¸ *response*.

Ðiи½Ñ Ð°ÐºÑ Ð¸Ñ :

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

headername â Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.
value ::= string | numeric | condition_name.

set

Ð ÐµÑ ÐµÐ¿Ð¸Ñ Ð°Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ ÐºÐ¾Ð½ÐºÑÐµÑ ÐµÑ Ð½Ð¾Ð¼Ñ
HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÑ . Ði Ð¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´Ð¸ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÑ Ñ
Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð² Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.
Ð ÐµÑ Ð²Ñ Ð¹ Ð¿Ð°Ñ Ñ Ð¼Ð¾Ð¶ÐµÑ Ñ Ð¾Ð¾Ð±Ñ, ÐµÑ Ð»Ð¸ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Ð¾Ñ Ð¾Ñ Ñ Ð¾ Ð³Ð¾Ð»Ð¾Ð²Ðµ *request* Ð¸Ð»Ð¸ *response*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

headername â Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.
value ::= string | numeric | condition_name.

delete

Ð£Ð´Ð°Ð»Ð¸Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº. ÐiÐ¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð² Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.

Ð ÐµÑ Ð²Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¼Ð¾Ð¶ÐµÑ Ð±Ñ Ñ Ñ Ð¾Ð¿Ñ Ñ ÐµÐ½, ÐµÑ Ð»Ð¸Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Ð¾Ñ Ð½Ð¾Ñ Ð¸Ñ Ñ Ñ Ðº Ð¾Ð´Ð½Ð¾Ð¹ Ð³Ñ Ñ Ð¿Ð¿Ðµ *request* Ð¸Ð»Ð¸ *response*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

headername â Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.

replace

Ð Ð¾Ð´Ð¸Ñ Ð¸Ñ Ð¸Ñ Ð¾Ð²Ð°Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ°. ÐiÐ¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð² Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.

Ð ÐµÑ Ð²Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¼Ð¾Ð¶ÐµÑ Ð±Ñ Ñ Ñ Ð¾Ð¿Ñ Ñ ÐµÐ½, ÐµÑ Ð»Ð¸Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Ð¾Ñ Ð½Ð¾Ñ Ð¸Ñ Ñ Ñ Ðº Ð¾Ð´Ð½Ð¾Ð¹ Ð³Ñ Ñ Ð¿Ð¿Ðµ *request* Ð¸Ð»Ð¸ *response*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

regex ::= string % Ñ ÐµÐ³Ñ Ð»Ñ Ñ Ð½Ð¾Ðµ Ð²Ñ Ñ Ð°Ð¶ÐµÐ½Ð¸Ðµ

value ::= string | condition_name

headername â Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.

Ð Ñ Ð¸Ð¼ÐµÑ 1:

Ð Ð¾Ð±Ð°Ð²Ð¸Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Referer:

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

Ð£Ð´Ð°Ð»Ð¸Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº:

```
PASS delete(Referer)
```

Ð ÐµÑ ÐµÐ¿Ð¸Ñ Ð°Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº:

Ð ÐµÐ¹Ñ Ñ Ð²Ð¸Ñ

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

Ð Ð¾Ð´Ð¸Ñ Ð¸Ñ Ð¾Ñ Ð¾Ð²Ð°Ñ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Location:

```
PASS response.header.Location.count = 1.. replace(response, Location, "http://example.com", url.ho
```

Ð Ñ Ð¸Ð¼ÐµÑ 2:

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

encrypt

Ð¨Ð¸Ñ Ñ Ð¾Ð²Ð°Ñ Ñ Ñ Ð°Ñ Ñ Ñ Ð¿Ñ Ñ Ð¸ Ð² HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐµ.
Ðið¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÑ Ñ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾ÐºÐ¾Ð² Ñ Ð¼. Ð²
Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.
Ð ÐµÑ Ð²Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¼Ð¾Ð¶ÐµÑ Ñ Ð¾Ð´Ð¸Ñ Ñ ÐµÐ½, ÐµÑ Ð»Ð¸ Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº Ð¾Ñ Ð½Ð¾Ñ Ð¸Ñ Ñ Ñ Ðº Ð¾Ð´Ð½Ð¾Ð¹ Ð³Ñ Ñ Ð¿Ð¿Ðµ *request* Ð¸Ð»Ð¸ *response*.
Ð Ð»Ñ Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ Ð¸ Ñ Ð»Ð°Ð³ "Ð Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ Ñ IP
ÐºÐ°Ðº Ñ Ð°Ñ Ñ Ñ ÐºÐ»Ñ Ñ Ð° Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ " â
Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ðµ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

url ::= string % Ñ Ð°Ñ Ñ Ñ url Ð´Ð»Ñ Ñ Ð¸Ð»Ñ Ñ Ñ Ð°Ñ Ð¸Ð¸

user_key ::= string % Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ñ ÐºÐ¸Ð¹ ÐºÐ»Ñ Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ (Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ)

add_ip ::= boolean % Ð´Ð¾Ð±Ð°Ð²Ð»Ñ Ñ Ñ Ð»Ð¸ IP Ðº ÐºÐ»Ñ Ñ Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ (Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¾Ð¼Ð¼ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ, Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ)

boolean ::= yes | no | true | false

headername â Ñ Ð¼. Ð² Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð¸.

encrypt_body_url

Ð¨Ð¸Ñ Ñ Ð¾Ð²Ð°Ñ Ñ Ñ Ð°Ñ Ñ Ñ Ð¿Ñ Ñ Ð¸ Ð² Ñ Ñ Ñ Ð»ÐºÐ°Ñ Ñ ÐµÐ»Ð°
Ð¾Ñ Ð²ÐµÑ Ð°.
Ð Ð»Ñ Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ Ð¸ Ñ Ð»Ð°Ð³ "Ð Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ Ñ IP
ÐºÐ°Ðº Ñ Ð°Ñ Ñ Ñ ÐºÐ»Ñ Ñ Ð° Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ " â
Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ðµ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

url ::= string % Ñ Ð°Ñ Ñ Ñ url Ð´Ð»Ñ Ñ Ð¸Ð»Ñ Ñ Ñ Ð°Ñ Ð¸Ð¸

user_key ::= string % Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ñ ÐºÐ¸Ð¹ ÐºÐ»Ñ Ñ Ñ Ð¾Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ (Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ)

add_ip ::= boolean % Ð´Ð¾Ð±Ð°Ð²Ð»Ñ Ñ Ñ Ð»Ñ IP ÐºÐ ÐºÐ»Ñ Ñ Ñ Ñ Ð¾Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ (Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¾Ð¼ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ, Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ)

boolean ::= yes | no | true | false

decrypt_path

Ð ÐµÑ Ð¸Ñ Ñ Ð¾Ð²Ð°Ñ Ñ Ñ Ð°Ñ Ñ Ñ Ð¿Ñ Ñ Ð¸ Ð·Ð°Ð¿Ñ Ð¾Ñ.

Ð ÐµÑ Ð²Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ Ð¼Ð¾Ð¶ÐµÑ Ð±Ñ Ð¾Ñ Ð¿Ñ Ð¾Ð²Ð»ÐµÐ½ Ð±Ñ Ð»Ð¾Ð³Ð¸Ñ Ñ ÐµÐ½Ð¸Ñ, ÐµÑ Ð»Ð¸ Ð¾Ð½Ð¸ Ð¾Ñ Ð¸Ð½Ð¸Ñ Ð¸Ñ Ñ Ñ Ðº Ð¾Ð´Ð½Ð¾Ð¹ Ð¾Ñ Ð¸Ð¸ *request* Ð¸Ð»Ð¸ *response*.

Ð Ð»Ñ Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ Ð¸ Ñ Ð»Ð°Ð³ "Ð Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ Ñ IP ÐºÐ°Ðº Ñ Ð°Ñ Ñ Ñ ÐºÐ»Ñ Ñ Ð° Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ " â Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ .

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

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

path ::= string % Ñ Ð°Ñ Ñ Ñ Ð¿Ñ Ñ Ð¸ Ð´Ð»Ñ Ñ Ð¸Ð»Ñ Ñ Ñ Ð°Ñ Ð¸Ð¸

user_key ::= string % Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ñ ÐºÐ¸Ð¹ ÐºÐ»Ñ Ñ Ñ Ð¾Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ (Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð½Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ)

add_ip ::= boolean % Ð´Ð¾Ð±Ð°Ð²Ð»Ñ Ñ Ñ Ð»Ñ IP ÐºÐ ÐºÐ»Ñ Ñ Ñ Ñ Ð¾Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ (Ð»Ð¾Ð³Ð¸Ñ ÐµÑ ÐºÐ¾Ð¼ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ, Ð½ÐµÐ¾Ð±Ñ Ð·Ð°Ñ ÐµÐ»Ñ Ð¹ Ð¿Ð°Ñ Ð°Ð¼ÐµÑ Ñ)

boolean ::= yes | no | true | false

Ð Ñ Ð¸Ð¼ÐµÑ :

Ð¨Ð¸Ñ Ñ Ð¾Ð²Ð°Ñ Ñ Ð²Ñ Ðµ Ð¾Ñ Ð½Ð¾Ñ Ð¸Ñ ÐµÐ»Ñ Ð½Ñ Ðµ Ð¿Ñ Ñ Ð¸ Ð² Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²Ð¾Ðº *Location* Ð¸ Ñ ÐµÐ»Ð¾Ðµ Ð¾Ñ Ð²ÐµÑ Ð°, Ð¸ Ð´ÐµÑ Ð¸Ñ Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ Ñ Ð¿Ñ Ð¾Ñ Ð¾Ð¼:

```
decrypt_path("/", "User_Key", true) enabled(true) name("Path decode")
http.response.code = 302 encrypt(Location, "/", "User_Key", true) enabled(true) name("Encrypt Loca
encrypt_body_url("/", "User_Key", true) enabled(true) name("Encrypt all relative URL")
```

body_inject

Ð Ñ Ñ Ð°Ð²Ð¸Ñ Ñ Ñ ÐºÑ Ð¸Ð¿Ñ Ð² Ñ ÐµÐ»Ð¾Ð¼ Ð¾Ñ Ð²ÐµÑ Ð°.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

body_inject(inject_text)

inject_text ::= string

set_cookie_token

Đ Đ¾Đ±Đ°Đ²Đ¸Ñ Ñ Đ² Đ¾Ñ Đ²ĐµÑ Đ·Đ°Đ³Đ¾Đ»Đ¾Đ²Đ¾Đº 'Set-Cookie' Ñ Đ¾Ñ Đ³ĐµĐ½ĐµÑ Đ¸Ñ Đ¾Đ²Đ°Đ½Đ½Ñ Đ¼ Ñ Đ¾ĐºĐµĐ½Đ¾Đ¼.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

set_cookie_token(cookie_name, parameter, expires_date)

cookie_name ::= string

parameter ::= string

expires_date ::= [DD_]HH:MM % Đ²Ñ ĐµĐ¼Ñ ĐºĐ¾Ñ Đ¾Ñ Đ¾Đµ Đ±Ñ Đ´ĐµÑ Đ¿Ñ Đ¸Đ±Đ°Đ²Đ»ĐµĐ½Đ¾ Đº Ñ ĐµĐºÑ Ñ ĐµĐ¼Ñ Đ²Ñ ĐµĐ¼ĐµĐ½Đ¸

Đ Ñ Đ¸Đ¼ĐµÑ :

Đ ĐµĐ°Đ»Đ¸Đ·Đ°Ñ Đ¸Ñ CSRF Đ·Đ°Ñ Đ¸Ñ Ñ :

```
DENY http.method = POST request.header.Referer.substring = "/login.php" qparam.UCSRF_TOKEN != requ
url.path.prefix = "/login.php" set_cookie_token(ucsrf_token, "path=/", 01_00:00) body_inject("<scr
    var tokenName = 'UCSRF_TOKEN';

    document.addEventListener('DOMContentLoaded', function()
    {
        var t_res = document.cookie.match(/ucsrf_token=(.+?)(;|$)/);
        var tokenValue = t_res ? t_res[1] : '';

        var forms = document.getElementsByTagName('form');
        for(i=0; i<forms.length; i++)
        {
            var html = forms[i].innerHTML;
            html += '<input type=hidden name=' + tokenName + ' value=' + tokenValue + ' />';
            forms[i].innerHTML = html;
        }
    });
</script>") enabled(true) name("Inject")
```

encode_cookie

Đ¨Đ¸Ñ Ñ Đ¾Đ²Đ°Ñ Ñ Đ·Đ½Đ°Ñ ĐµĐ½Đ¸Ñ Cookie Đ² Đ·Đ°Đ³Đ¾Đ»Đ¾Đ²ĐºĐµ *Set-Cookie* Ñ Đ·Đ°Đ´Đ°Đ½Đ½Ñ Đ¼ Đ¸Đ¼ĐµĐ½ĐµĐ¼.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

cookie_name ::= string

condition_name % Ñ Ñ Đ»Đ¾Đ²Đ¸Đµ Đ¸Ñ Đ¿Đ¾Đ»Ñ Đ·Ñ ĐµĐ¼Đ¾Đµ Đ´Đ»Ñ ĐºĐ¾Đ´Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ src.ip)

user_kry_string ::= string % Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ Ñ ĐºĐ¸Đ ĐºĐ»Ñ Ñ Đ¸Ñ Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ "")

f_encrypt := true % Đ½ĐµĐ¾Đ±Ñ Đ¾Đ´Đ¸Đ¼Đ¾ Ñ Đ¸Ñ Ñ Đ¾Đ²Đ°Đ½Đ¸Đµ (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ false)

decode_cookie

Đ ĐµÑ Đ¸Ñ Ñ Đ¾Đ²Đ°Ñ Ñ Đ¾ĐºĐµĐ½ Đ² Đ·Đ°Đ³Đ¾Đ»Đ¾Đ²ĐºĐµ Cookie Ñ Đ·Đ°Đ´Đ°Đ½Đ½Ñ Đ¼ Đ¸Đ¼ĐµĐ½ĐµĐ¼.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

cookie_name ::= string

condition_name % Ñ Ñ Đ»Đ¾Đ²Đ¸Đµ Đ¸Ñ Đ¿Đ¾Đ»Ñ Đ·Ñ ĐµĐ¼Đ¾Đµ Đ´Đ»Ñ Đ´ĐµĐºĐ¾Đ´Đ¸Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ src.ip)

user_kry_string ::= string % Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ Ñ ĐºĐ¸Đ ĐºĐ»Ñ Ñ Đ¸Ñ Ñ Đ¾Đ²Đ°Đ½Đ¸Ñ (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ "")

f_encrypt := true % Đ½ĐµĐ¾Đ±Ñ Đ¾Đ´Đ¸Đ¼Đ¾ Ñ Đ¸Ñ Ñ Đ¾Đ²Đ°Đ½Đ¸Đµ (Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ false)

Đ Ñ Đ¸Đ¼ĐµÑ :

Đ¨Đ¸Ñ Ñ Đ¾Đ²Đ°Đ½Đ¸Đµ Đ¸ Đ´ĐµÑ Đ¸Ñ Ñ Đ¾Đ²Đ°Đ½Đ¸Đµ Cookie Ñ Đ¸Đ¼ĐµĐ½ĐµĐ¼ *security:*

```
response.header.Set-Cookie.count != 0 encode_cookie("security", src.ip, true) enabled(true) name("
request.header.Cookie.count != 0 decode_cookie("security", src.ip, true) enabled(true) name("decoc
```

body_replace

Đ Đ¾Đ´Đ¸Ñ Đ¸Ñ Đ¾Đ²Đ°Ñ Ñ Ñ ĐµĐ»Đ¾ Đ¾Ñ Đ²ĐµÑ Đ°. Đ Ñ Đ¿Đ¾Đ»Đ½Ñ ĐµÑ Ñ Ñ Đ½Đµ Đ±Đ¾Đ»ĐµĐµ Đ´Đ²Ñ Ñ (Đ¿ĐµÑ Đ²Ñ Ñ) Đ¼Đ¾Đ´Đ¸Ñ Đ¸ĐºÐ°Ñ Đ¸Đ¹ Đ´Đ»Ñ ĐºÐ°Đ¶Đ´Đ¾Đ³Đ¾ Đ¾Ñ Đ²ĐµÑ Đ°.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

body_replace(<regex>, <value>)

regex ::= string % Ñ ĐµĐ³Ñ Đ»Ñ Ñ Đ½Đ¾Đµ Đ²Ñ Ñ Đ°Đ¶ĐµĐ½Đ¸Đµ

value ::= string

Đ Ñ Đ¸Đ¼ĐµÑ :

```
PASS \
body_replace("(\\+7|8)[\\s(]?(\\d\{3})[\\s)]?(\\d\{3})[\\s-]?(\\d\{2})[\\s-]?(\\d\{2})", "+\\1 (\\
body_replace("(\\w{1})[\\w\.]*(\\w{1})@([\\w]+)\\.([\\w]+)", "\\1***\\2@\\3.\\4") \
enabled(true) \
name("Replace mail and phone")
```

lookup_and_auth

Đ Ñ Ñ ĐµĐ½Ñ Đ¸Ñ Đ¸Ñ Đ¾Đ²Đ°Ñ Ñ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ . Đ Ñ Đ»Ñ Ñ Đ°Đµ ĐµÑ Đ»Đ¸ IP Đ½Đµ Ñ ĐºĐ°Đ·Đ°Đ½, Đ·Đ°Đ¿Ñ Đ¾Ñ Đ¼Đ°Ñ ĐºĐ¸Ñ Ñ ĐµÑ Ñ Ñ Đ¸Đ¼ĐµĐ½Đ¸ĐµĐ¼ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ .

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

user_login ::= string | condition_name % Đ Đ¾Đ³Đ¸Đ½ Đ°Ñ Ñ ĐµĐ½Ñ Đ¸Ñ Đ¸ĐºĐ°Ñ Đ¸Đ

ip_address ::= string | condition_name % IP Đ°Đ´Ñ ĐµÑ

session_timeout ::= integer % Ñ Đ°Đ¹Đ¼-Đ°Ñ Ñ Ñ ĐµÑ Ñ Đ¸Đ¸, Đ¿Đ¾ Ñ Đ¼Đ¾Đ»Ñ Đ°Đ½Đ¸Ñ 0.

Đ Ñ Đ¸Đ¼ĐµÑ :

```
lookup_and_auth(request.x_header.X-Authenticated-User, request.x_header.X-Forwarded-For, 300) enab
lookup_and_auth(request.x_header.X-Authenticated-User) enabled(true) name("Mark request")
```

redirect

Đ Ñ Đ¸ Đ±Đ¾Đ¼ĐºĐ¸Ñ Đ¾Đ²ĐºĐµ Đ¿ĐµÑ ĐµĐ½Đ°Đ¿Ñ Đ°Đ²Đ¸Ñ Ñ Đ¿Đ¾Đ»Ñ Đ·Đ¾Đ²Đ°Ñ ĐµĐ»Ñ Đ½Đ° Đ°Đ´Ñ ĐµÑ , ĐºĐ¾Ñ Đ¾Ñ Ñ Đ¹ Ñ ĐºĐ°Đ·Đ°Đ½ Đ² Ñ ĐµĐ´Đ¸Ñ ĐµĐºÑ Đµ.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

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

RespCode ::= 301 | 302 | 305 | 307

RedirectText ::= string

Url ::= string

Đ Ñ Đ¸Đ¼ĐµÑ :

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

inc Đ¸ dec

Đ Ñ Đ¿Đ¾Đ»Ñ Đ·Ñ Ñ Ñ Ñ Ñ Đ´Đ»Ñ Đ¸Đ·Đ¼ĐµÑ½ĐµÑ½Đ¸Ñ Đ·Đ½Đ°Ñ ĐµÑ½Đ¸Ñ Đ¿ĐµÑ ĐµÑ¼ĐµÑ½Đ½Ñ Ñ , Đ¾Đ±Ñ Ñ Đ²Đ»ĐµÑ½Đ½Ñ Ñ ĐºĐ°Đº *def var*.

ĐiĐ¸Đ½Ñ Đ°ĐºÑ Đ¸Ñ :

inc(var.<var_name>, integer)

dec(var.<var_name>, integer)

Ð Ñ Ð¸Ð¼ÐµÑ :

Ð Ð° ÐºÐ°Ð¶Ð´Ñ Ð¹ *http.response.code = 500* Ñ Ð²ÐµÐ»Ð¸Ñ Ð¸Ð²Ð°ÐµÑ Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ *rps* Ð½Ð° 1. Ð Ñ Ð»Ð¸ Ð¿Ñ ÐµÐ²Ñ Ñ Ð¸Ð»Ð¸ 10 Ñ Ð°ÐºÐ¸Ñ Ð·Ð°Ð¿Ñ Ð¾Ñ Ð¾Ð² Ð·Ð° 5 Ð¼Ð¸Ð½Ñ Ñ , Ð±Ð»Ð¾ÐºÐ¸Ñ Ñ ÐµÐ¼ Ð´Ð°Ð»Ñ Ð½ÐµÐ¹Ñ Ð¸Ðµ Ð¾Ñ Ð²ÐµÑ Ñ . Ð§ÐµÑ ÐµÐ· 5 Ð¼Ð¸Ð½Ñ Ñ Ð¿ÐµÑ ÐµÐ¼ÐµÐ½Ð½Ð°Ñ *rps* Ð±Ñ Ð´ÐµÑ Ñ Ð±Ñ Ð¾Ñ ÐµÐ½Ð° Ð² 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

ÐiÐ±Ñ Ð¾Ñ Ð¸Ñ Ñ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ð¿ÐµÑ ÐµÐ¼ÐµÐ½Ð½Ñ Ñ , Ð¾Ð±Ñ Ñ Ð²Ð»ÐµÐ½Ð½Ñ Ñ ÐºÐ°Ðº *init* Ð² *def var,* Ð² Ð½Ð°Ñ Ð°Ð»Ñ Ð½Ð¾Ðµ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ðµ.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

reset(var.<var_name>)

sma

Ð Ñ Ð¿Ð¾Ð»Ñ Ð·Ñ Ñ Ñ Ñ Ñ Ð´Ð»Ñ Ð¿Ð¾Ð´Ñ Ñ ÐµÑ Ð° Ñ Ñ ÐµÐ´Ð½ÐµÐ³Ð¾ Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ð² Ð¾ÐºÐ½Ðµ Ð²Ñ ÐµÐ¼ÐµÐ½Ð¸, ÐºÐ¾Ñ Ð¾Ñ Ð¾Ðµ Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»Ñ ÐµÑ Ñ Ñ Ð² Ð¿ÐµÑ ÐµÐ¼ÐµÐ½Ð½Ð¾Ð¹ ÐºÐ°Ðº *window* Ð² *def var*.

ÐiÐ¸Ð½Ñ Ð°ÐºÑ Ð¸Ñ :

sma(var.<var_name>, integer)

Ð Ñ Ð¸Ð¼ÐµÑ :

Ð Ð»Ð¾ÐºÐ¸Ñ Ñ Ñ Ñ Ñ Ñ Ð·Ð°Ð¿Ñ Ð¾Ñ Ñ , ÐºÐ¾Ð³Ð´Ð° Ñ Ñ ÐµÐ´Ð½ÐµÐµ Ð²Ñ ÐµÐ¼Ñ Ð·Ð°Ð¿Ñ Ð¾Ñ Ð° Ð·Ð° 30-Ñ ÐµÐºÑ Ð½Ð´Ð½Ñ Ð¹ Ð¸Ð½Ñ ÐµÑ Ð²Ð°Ð» Ð¿Ñ ÐµÐ²Ñ Ñ Ð¸Ñ 2 Ñ ÐµÐºÑ Ð½Ð´Ñ :

```
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 â Ð¿Ð¾Ð´Ð¼ÐµÐ½Ð° Ñ ÐµÑ ÐµÐ²Ñ Ñ IP-Ð°Ð´Ñ ÐµÑ Ð¾Ð².

Ð Ñ Ð¸Ð¼ÐµÑ :

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

dnat

DNAT â Ð¿Ð¾Ð´Ð¼ÐµÐ½Ð° IP-Ð°Ð´Ñ ÐµÑ Ð° Ð½Ð°Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ .

Ð Ñ Ð¸Ð¼ÐµÑ :

```
PASS dnat target_ip("192.168.1.20")
```

port_mapping

Ð Ð¾Ñ Ñ -Ñ Ð¾Ñ Ð²Ð°Ñ Ð´Ð¸Ð½Ð³ â Ð¿ÐµÑ ÐµÐ½Ð°Ð¿Ñ Ð°Ð²Ð»ÐµÐ½Ð¸Ñ Ñ Ð°Ñ Ð¸ÐºÐ° Ð½Ð° Ñ ÐºÐ°Ð·Ð°Ð½Ð½Ñ Ð¹ IP-Ð°Ð´Ñ ÐµÑ c
Ð¸Ð·Ð¼ÐµÐ½ÐµÐ½Ð¸Ð¼ Ð½Ð¾Ð¼ÐµÑ Ð° Ð¿Ð¾Ñ Ñ Ð°
Ð¿Ñ Ð±Ð»Ð¸ÐºÐ¾Ð²Ð°Ð½Ð½ Ñ ÐµÑ Ð²Ð¸ Ð°.

Ð Ñ Ð¸Ð¼ÐµÑ :

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

netmap

Network mapping â Ð¿Ð¾Ð·Ð²Ð¾Ð»Ñ ÐµÑ Ð¾Ð¸Ð·Ð²ÐµÑ Ñ Ð¸ Ð·Ð°Ð¼ÐµÐ½Ñ IP-Ð°Ð´Ñ ÐµÑ Ð¾Ð² Ð¸Ñ Ñ Ð¾Ñ Ð½Ð¸Ð°Ð° Ð¸Ð»Ð¸ Ð½Ð°Ð·Ð½Ð°Ñ ÐµÐ½Ð¸Ñ Ð¾Ð´Ð½Ð¾Ð¹ Ñ ÐµÑ Ð¸ Ð½Ð° Ð´Ñ Ñ Ð³Ñ Ñ .

Ð Ñ Ð¸Ð¼ÐµÑ :

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

route

Policy-based routing â Ð¿Ð¾Ð·Ð²Ð¾Ð»Ñ ÐµÑ Ð¼Ð°Ñ Ñ Ñ Ñ Ð¸Ð·Ð¸Ñ Ð¾Ð²Ð°Ñ Ñ IP-Ð¿Ð°ÐºÐµÑ Ñ Ð½Ð° Ð¾Ñ Ð½Ð¾Ð²Ðµ Ñ Ð°Ñ Ñ Ð¸ ÐµÐ½Ð½Ð¾Ð¹
Ð¸Ð½Ñ Ð¾Ñ Ð¼Ð°Ñ Ð¸, Ð½Ð°Ð¿Ñ Ð¸Ð¼ÐµÑ , Ñ ÐµÑ Ð²Ð¸ Ð¾Ñ Ð²,
MAC-Ð°Ð´Ñ ÐµÑ Ð¾Ð² Ð¸Ð»Ð¸ Ñ ÐµÑ Ð²ÐµÑ Ð¾Ð² (IP-Ð°Ð´Ñ ÐµÑ Ð¾Ð²).
SNAT IP â Ð Ð´Ñ ÐµÑ , Ð½Ð° ÐºÐ¾Ñ Ð¾Ñ Ñ Ð¹ Ð±Ñ Ð´ÐµÑ Ð·Ð°ÐºÐ¾ÐµÐ½ÐµÐ½ Ð°Ð´Ñ ÐµÑ Ð¸Ñ Ñ Ð¾Ñ Ð½Ð¸Ð°Ð° Ð´Ð»Ñ Ñ Ñ Ð°Ñ Ð¸ÐºÐ° NAT.

Ð Ñ Ð¸Ð¼ÐµÑ :

```
PASS service = HTTP route getway(Ð¨Ð»Ñ Ð·1)
```

ignore

Ð Ð³Ð½Ð¾Ñ Ð¸Ñ Ð¾Ð²Ð°Ñ Ñ Ð¾Ñ Ð²ÐµÑ Ð¾Ñ ICAP-Ñ ÐµÑ Ð²ÐµÑ Ð°. Ð Ñ Ñ Ð¾Ð¼ Ñ Ð»Ñ Ñ Ð°Ðµ, Ð²Ð½Ðµ Ð·Ð°Ð²Ð¸Ñ Ð¸Ð¼Ð¾Ñ Ñ Ð¸ Ð¾Ñ Ð¾Ñ Ð²ÐµÑ Ð° ICAP-Ñ ÐµÑ Ð²ÐµÑ Ð°, Ð´Ð°Ð½Ð½Ñ Ðµ Ðº Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ñ Ñ Ð¾Ð´Ñ Ñ Ð±ÐµÐ· Ð¼Ð¾Ð´Ð¸Ñ Ð¸ÐºÐ°Ñ Ð¸Ð¸.

Ð Ñ Ð¸Ð¼ÐµÑ :

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

forward

Ð ÐµÑ ÐµÑ Ð»Ð°Ñ Ñ . Ð Ñ Ð»Ñ Ñ Ð°Ðµ Ñ Ñ Ð¿ÐµÑ Ð½Ð¾Ð¹ Ñ Ð°Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²ÐºÐ¸ Ñ Ñ Ð°Ñ Ð¸ÐºÐ° SSL/TLS ÐºÐ¾Ð¿Ð¸Ñ Ñ Ñ Ð°Ñ Ð¸ÐºÐ° Ð±Ñ Ð´ÐµÑ Ð¿ÐµÑ ÐµÑ Ð»Ð°Ð½Ð° Ð² Ñ Ð¾Ð¾Ñ Ð²ÐµÑ Ñ Ñ Ð²Ð¸Ð¸ Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾Ð¼ Ð¸ Ð¿Ñ Ð¾Ñ Ð¸Ð»ÐµÐ¼ Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL.

Ð Ñ Ð¸Ð¼ÐµÑ :

```
OK url = lib.url(ZAPRET_INFO_BLACK_LIST_DOMAIN)ssl_profile("Default SSL profile") ssl_forward_prof
```

Ð¢Ð¸Ð¿Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»

Version: 7.x
Product: NGFW

Ð Ñ Ð°Ð²Ð¸Ð»Ð° ÐºÐ¾Ð½Ñ ÐµÐ½Ñ Ð½Ð¾Ð¹ Ñ Ð¸Ð»Ñ Ñ Ñ Ð°Ñ Ð¸Ð¸

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ð°Ð·Ñ ÐµÑ ÐµÐ½Ð¸Ðµ Ð¿Ð¾Ñ ÐµÑ ÐµÐ½Ð¸Ñ .
DENY	Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ð²ÐµÐ±-Ñ Ñ Ñ Ð°Ð½Ð¸Ñ Ñ .
WARNING	Ð Ñ ÐµÐ´Ñ Ð¿Ñ ÐµÐ¶Ð´ÐµÐ½Ð¸Ðµ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ð¾ Ñ Ð¾Ð¼, Ñ Ñ Ð¾ Ñ Ñ Ñ Ð°Ð½Ð¸Ñ Ð° Ð½ÐµÐ¶ÐµÐ»Ð°Ñ ÐµÐ»Ñ Ð½Ð° Ð´Ð»Ñ Ð¿Ð¾Ñ ÐµÑ ÐµÐ½Ð¸Ñ .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

category, morphology, scenario, time, url, user.

request.header.Referer â ÐiÐ¿Ð¸Ñ Ð¾Ðº URL, Ð² ÐºÐ¾Ñ Ð¾Ñ Ð¾Ð¼ Ñ ÐºÐ°Ð·Ð°Ð½Ñ Ñ ÐµÑ ÐµÑ ÐµÑ Ñ Ð´Ð»Ñ Ñ ÐµÐºÑ Ñ ÐµÐ¹ Ñ Ñ Ñ Ð°Ð½Ð¸Ñ Ñ .
request.header.User-Agent â Useragent Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ñ ÐµÐ»Ñ Ñ ÐºÐ¸Ñ Ð±Ñ Ð°Ñ Ð·ÐµÑ Ð¾Ð².
response.header.Content-Type â ÐiÐ¿Ð¸Ñ ÐºÐ¸ Ñ Ð¸Ð¿Ð¾Ð² ÐºÐ¾Ð½Ñ ÐµÐ½Ñ Ð°.

http.method â Ð ÐµÑ Ð¾Ð´, Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ñ ÐµÐ¼Ñ Ð¹ Ð²
HTTP-Ð·Ð°Ð¿Ñ Ð¾Ñ Ð°Ñ .

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, virus_heuristic.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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
    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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð¼ÐµÐ¶Ñ ÐµÐ²Ð¾Ð³Ð¾ Ñ ÐºÑ Ð°Ð½Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ð°Ð·Ñ ÐµÑ ÐµÐ½Ð¸Ðµ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.
DENY	Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²Ð²ÐºÐ° Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

service, scenario, time, url, user, hip_profile.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, reject_with, ips_profile, l7_profile.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[firewall "Firewall rules"]
% ----------------- 1 -----------------
DENY \
    scenario = "Example torrent detection scenario" \
    dst.zone = Untrusted \
    dst.ip = lib.network("Botnets IP list") \
    rule_log(session) \
    reject_with("host-unreach") \
    enabled(true) \
    name("Example block RU RKN by IP list")
% ----------------- 2 -----------------
PASS \
    scenario = "Example torrent detection scenario" \
    src.zone = Trusted \
    dst.zone = Untrusted \
    hip_profile = "HIP profile" \
    ips_profile("Default IDPS profile") \
    l7_profile("Pass all applications") \
    rule_log(yes, "3/h", 5) \
    enabled(true) \
    name("Allow trusted to untrusted")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° NAT Ð¸ Ð¼Ð°Ñ Ñ Ñ Ñ Ð¸Ð·Ð°Ñ Ð¸Ð

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð¢Ð¸Ð¿ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð° Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»Ñ ÐµÑ Ñ Ð´ÐµÐ¹Ñ Ð²Ð¸ÐµÐ¼ (Action):

- nat â Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ NAT;
- dnat â Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ DNAT;
- port_mapping â Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Ð¿Ð¾Ñ Ñ -Ñ Ð¾Ñ Ð²Ð°Ñ Ð´Ð¸Ð½Ð³Ð°;
- netmap â Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ Network mapping;
- route â Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾ policy-based.

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

service, scenario, time, url, user.

Ðiв²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, direction, target_ip, target_snat, snat_target_ip,

port_map, gateway.

Ð Ñ Ð¸ÐµÑ

```
[nat_routing "NAT and Routing Rules"]
% ---------------- 1 ----------------
OK \
    src.zone = Trusted \
    service = (HTTP, HTTPS) \
    snat_target_ip("192.168.13.210") \
    rule_log(session) \
    enabled(true) \
    name("NAT Rule")\
    nat
% ---------------- 2 ----------------
OK \
    src.zone = Management \
    dst.ip = (lib.network("Private IPs"), lib.url("Microsoft Windows Internet checker")) \
    target_ip("171.168.1.1") \
    target_snat(yes) \
    snat_target_ip("192.168.1.1") \
    enabled(true) \
    name("DNAY Rule")\
    dnat
% ---------------- 3 ----------------
OK \
    target_ip("172.168.1.1") \
    snat_target_ip("192.168.1.1") \
    port_map(tcp, 2000, 2000) \
    enabled(true) \
    name("Port-forwarding Rule")\
    port_mapping
% ---------------- 4 ----------------
OK \
    user = example \
    scenario = "Example torrent detection scenario" \
    gateway(My) \
    enabled(true) \
    name("Policy-base Rule")\
    route
% ---------------- 5 ----------------
OK \
    dst.geoip = (RW, SO) \
    target_ip("172.168.1.1") \
    direction(input) \
    enabled(true) \
    name("Network mapping Rule")\
    netmap
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Captive-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ðµ Ð¸Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ð½Ñ Ñ Ð°Ñ Ñ ÐµÐ½Ð¸Ð¸ Ð¸Ñ Ð¿ÐºÐ°Ð¸Ñ .
OK	Ð Ñ Ð¿Ð¾Ð»Ñ Ð·Ð¾Ð²Ð°Ð½Ñ Ñ Captive-Ð¿Ñ Ð¾Ñ Ð¸Ð»Ñ .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

time, url, category.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, profile.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ðµ Ñ Ð°Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ñ Ð²Ð°Ñ Ñ Ð¿ÐµÑ ÐµÐ´Ð°Ð²Ð°ÐµÐ¼Ñ Ðµ Ð´Ð°Ð½Ð½Ñ Ðµ.
OK	Ð Ð°Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ñ Ñ Ð¿ÐµÑ ÐµÐ´Ð°Ð²Ð°ÐµÐ¼Ñ Ðµ Ð´Ð°Ð½Ð½Ñ Ðµ.

Ð ÐµÑ ÐµÑ Ñ Ð»ÐºÐ° Ð¾Ð¿Ñ ÐµÐ´ÐµÐ»Ñ ÐµÑ Ñ Ñ Ð´ÐµÐ¹Ñ Ñ Ð²Ð¸ÐµÐ¼ (Action):

- forward â Ð² Ñ Ð»Ñ Ñ Ð°Ðµ Ñ Ñ Ð¿ÐµÑ Ð½Ð¾Ð¹ Ñ Ð°Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²ÐºÐ¸Ñ Ñ Ð°Ñ Ð¸ÐºÐ° SSL/TLS ÐºÐ¾Ð¿Ð¸Ñ Ñ Ñ Ð°Ñ Ð¸ÐºÐ° Ð±Ñ Ð´ÐµÑ Ð¿ÐµÑ ÐµÑ Ð»Ð°Ð½Ð° Ð² Ñ Ð¾Ð¾Ñ Ð²ÐµÑ Ñ Ñ Ð²Ð¸Ð¸ Ñ Ð¿Ñ Ð°Ð²Ð¸Ð»Ð¾Ð¼Ð¸ Ð¿Ñ Ð¾Ñ Ð¸Ð»ÐµÐ¼ Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSL (ssl_forward_profile).

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip,

dst.geoip, dst.ip,

time, service, user, category.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, ssl_profile, ssl_forward_profile.

block_invalid_cert, check_revoc_cert, block_expired_cert, block_self_signed_cert.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ SSH

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ðµ Ñ Ð°Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ñ Ð²Ð°Ñ Ñ Ð¿ÐµÑ ÐµÐ´Ð°Ð²Ð°ÐµÐ¼Ñ Ðµ Ð´Ð°Ð½Ð½Ñ Ðµ.
OK	Ð Ð°Ñ Ñ Ð¸Ñ Ñ Ð¾Ð²Ð°Ñ Ñ Ð¿ÐµÑ ÐµÐ´Ð°Ð²Ð°ÐµÐ¼Ñ Ðµ Ð´Ð°Ð½Ð½Ñ Ðµ.

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip,

dst.geoip, dst.ip,

time, service, user.

Ð¡Ð²Ð¾Ð¹ Ñ Ð²Ð°

name, desc, enabled, rule_log,

block_ssh_shell, block_ssh_exec, block_sftp, ssh_command.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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-Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° OK.

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

url.domain.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, dns_server.

Ð Ñ Ð¸Ð¼ÐµÑ

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

DoS-Ð¿Ñ Ð°Ð²Ð¸Ð»Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ð°Ð·Ñ ÐµÑ ÐµÐ½Ð¸Ðµ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.
DENY	Ð ÐµÐ·Ñ Ñ Ð»Ð¾Ð²Ð½Ð°Ñ Ð±Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ñ Ñ Ð°Ñ Ð¸ÐºÐ°.

Đ Đ¼Ñ	Đ Đ¿Đ¸Ñ Đ°Đ½Đ¸Đµ
WARNING	Đ Ñ Đ¸Đ¼ĐµĐ½ĐµĐ½Đ¸Đµ Đ¿Ñ Đ¾Ñ Đ¸Đ»Ñ Đ·Đ°Ñ Đ¸Ñ Ñ Đ¾Ñ DoS-Đ°Ñ Đ°Đº.

Đ£Ñ Đ»Đ¾Đ²Đ¸Ñ

src.zone, src.geoip, src.ip,

dst.zone, dst.geoip, dst.ip.

time, service, user, scenario,

Đ¡Đ²Đ¾Đ¹Ñ Ñ Đ²Đ°

name, desc, enabled, rule_log, profile.

Đ Ñ Đ¸Đ¼ĐµÑ

```
[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-Đ¿Ñ Đ°Đ²Đ¸Đ»Đ°

Đ Ñ ĐµÑ Đ¸ĐºÑ Ñ

Đ Đ¼Ñ	Đ Đ¿Đ¸Ñ Đ°Đ½Đ¸Đµ
PASS	Đ Ñ Đ¾Đ¿Ñ Ñ Đº. Đ Đµ Đ¿Đ¾Ñ Ñ Đ»Đ°Ñ Ñ Đ´Đ°Đ½Đ½Ñ Đµ Đ½Đ° ICAP-Ñ ĐµÑ Đ²ĐµÑ .
OK	Đ ĐµÑ ĐµÑ Ñ Đ»ĐºĐ° Đ´Đ°Đ½Đ½Ñ Ñ Đ½Đ° ICAP-Ñ ĐµÑ Đ²ĐµÑ .

Đ ĐµĐ¹Ñ Ñ Đ²Đ¸Đ (Action):

- ignore â Đ¸Đ³Đ½Đ¾Ñ Đ¸Ñ Đ¾Đ²Đ°Ñ Ñ Đ¾Ñ Đ²ĐµÑ Đ¾Ñ ICAP-Ñ ĐµÑ Đ²ĐµÑ Đ°.

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

url, category, user, service, http.method, response.header.Content-Type.

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, profile.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð·Ð°Ñ Ñ Ð¿Ð¾Ñ Ñ Ð¾Ð²Ð¾Ð³Ð¾ Ñ Ñ Ð°Ñ Ð¸ÐºÐ°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
PASS	Ð Ñ Ð¾Ð¿Ñ Ñ Ðº Ñ Ñ Ð°Ñ Ð¸ÐºÐ° Ð±ÐµÐ· Ð¸Ð·Ð¼ÐµÐ½ÐµÐ½Ð¸Ð¹.
DENY ("with error")	Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ð¿Ð¸Ñ Ñ Ð¼Ð°, Ð¿Ñ Ð¸ Ñ Ñ Ð¾Ð¼ Ñ Ð¾Ð¾Ð±Ñ Ð°ÐµÑ Ñ Ð¾Ð¾Ð± Ð¾Ñ Ð¸Ð±ÐºÐµ Ð´Ð¾Ñ Ñ Ð°Ð²ÐºÐ¸ Ð¿Ð¸Ñ Ñ Ð¼Ð° Ð½Ð° Ñ ÐµÑ Ð²ÐµÑ .
DENY	Ð Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ð¿Ð¸Ñ Ñ Ð¼Ð° Ð±ÐµÐ· Ñ Ð²ÐµÐ´Ð¾Ð¼Ð»ÐµÐ½Ð¸Ñ Ð¾ Ð±Ð»Ð¾ÐºÐ¸Ñ Ð¾Ð²ÐºÐµ.
WARNING	Ð Ð°Ñ ÐºÐ¸Ñ Ð¾Ð²ÐºÐ° Ð¿Ð¾Ñ Ñ Ð¾Ð²Ñ Ñ Ð¾Ð¾Ð±Ñ ÐµÐ½Ð¸Ñ.

Đ£Ñ Đ»Đ¾Đ²Đ¸Ñ

src.zone, src.geoip, src.ip.

dst.zone, dst.geoip, dst.ip.

user, service, envelope_from, envelope_to.

ĐiĐ²Đ¾Đ¹Ñ Ñ Đ²Đ°

name, desc, enabled, rule_log, mark_hdr, mark, antispam_usergate, dnsbl.

Đ Ñ Đ¸Đ¼ĐµÑ

```
[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")
```

Đ Ñ Đ°Đ²Đ¸Đ»Đ° reverse-Đ¿Ñ Đ¾ĐºÑ Đ¸

Đ Ñ ĐµÑ Đ¸ĐºÑ Ñ

Đ Đ¼Ñ	Đ Đ¿Đ¸Ñ Đ°Đ½Đ¸Đµ
OK	Đ Ñ ĐµÐ³Đ´Đ° Đ Đ .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

user, request.header.User-Agent, url.port.

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, profile, certificate, cert_auth_enabled, is_https, ssl_profile,

rewrite_path.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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 r
    profile("Example reverse proxy server") \
    rewrite_path("example.com/path1", "example.local/path1") \
    enabled(true) \
    name("Example reverse proxy rule")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð²ÐµÐ±-Ð±ÐµÐ·Ð¾Ð¿Ð°ÑÐ½Ð¾ÑÑ Ð¸

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

time, user,

ÐiÐ²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log.

enable_adblock, safe_search, search_history_logging, social_sites_block, enable_injector,

custom_injector, url_list_exclusions.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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 deletec
    enabled(false) \
    name("Example safebrowsing")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð¿Ñ Ð¾Ð¿Ð¸Ñ Ñ ÐºÐ½Ð¾Ð¿Ð¾Ðº Ñ Ð¿Ð¾Ñ Ð¾Ð±Ð½Ð¾Ñ Ñ Ð¸

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

time, user application, service, scenario,

Ði Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, rule_log, bandwidth_pool.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð¸Ð½Ñ Ð¿ÐµÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ Ñ Ð½Ð½ÐµÐ»ÐµÐ¹

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

dst.zone, dst.geoip, dst.ip.

service.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled.

Ð Ñ Ð¸Ð¼ÐµÑ

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

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð²ÐµÐ±-Ð¿Ð¾Ñ Ñ Ð°Ð»Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

url, user, url.domain.

Ð¡Ð²Ð¾Ð¹ÑÑ Ñ Ð²Ð°

name, desc, enabled.

icon, ssl_profile, certificate, additional_url, rdp_check_session_alive, transparent_auth.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° VPN-Ñ ÐµÑ Ð²ÐµÑ Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

dst.geoip, dst.ip.

user,

Ð¡Ð²Ð¾Ð¹ÑÑ Ñ Ð²Ð°

name, desc, enabled.

profile, auth_profile, vpn_network, interface.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° VPN-ÐºÐ»Ð¸ÐµÐ½Ñ Ð°

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ði Ð²Ð¾Ð¹Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled.

profile, interface, server_address.

last_error, status â Ð¸Ð½Ñ Ð¾Ñ Ð¼Ð°Ñ Ð¸Ð½Ð²Ð¾Ð¼ Ð¿Ð¾Ð»Ðµ, Ð½ÐµÐ´Ð¾Ñ Ñ Ð¿Ð½Ð¾Ðµ Ð´Ð»Ñ ÐµÐ´Ð°ÐºÑ Ð¸Ñ Ð¾Ð²Ð°Ð½Ð¸Ñ .

Ð Ñ Ð¸Ð¼ÐµÑ

```
[vpn_client "VPN client Rules"]
% ----------------- 1 -----------------
OK \
    server_address("10.10.10.10") \
    last_error(Disabled) \
    status(disconnected) \
    profile("Client VPN profile") \
    interface(tunnel3) \
    enabled(true) \
    name("Client VPN rule")
```

Ð Ñ Ð°Ð²Ð¸Ð» Ñ Ñ ÐµÐ½Ñ Ð¸ÐµÐ²

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

scenario_cond.

Ði Ð²Ð¾Ð¹Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled.

operation_mode, trigger, duration.

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ Ñ Ñ ÐµÐ½Ð°Ñ Ð¸Ñ

Ð Ð°Ñ ÐµÐ³Ð¾Ñ Ð¸Ñ URL (url_category)

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

category

Ði Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

count_interval, max_event_count, scond_type

Ð Ð±Ð½Ð°Ñ Ñ Ð¶ÐµÐ½Ð¸Ð Ð²Ð¸Ñ Ñ Ñ Ð° (virus_detection)

Ði Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

scond_type

Ð Ñ Ð¸Ð»Ð¾Ð¶ÐµÐ½Ð¸Ð (app)

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

application

Ði Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

count_interval, max_event_count, scond_type

Ði Ð (ips)

Ði Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

ips_tl, scond_type

Ð¢Ð¸Ð¿ ÐºÐ¾Ñ ÐµÐ½Ð° (mime_type)

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

response.header.Content-Type

Ði Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

count_interval, max_event_count, scond_type

Ð Ð°Ð·Ð¼ÐµÑ Ð¿Ð°ÐºÐµÑ Ð° (net_packet_size)

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

packet_size, scond_type

Ð¡ÐµÑ Ñ Ð¸Ð¹ c Ð¾Ð´Ð½Ð¾Ð³Ð¾ IP (sessions_per_ip)

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

sessions_limit, scond_type

Ð Ð±Ñ ÐµÐ¼ Ñ Ñ Ð°Ñ Ð¸ÐºÐ° (traffic)

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

traffic_limit, period, scond_type

Ð Ñ Ð¾Ð²ÐµÑ ÐºÐ° Ñ Ð¾Ñ Ñ Ð¾Ñ Ð½Ð¸ (health_check)

Ð£Ñ Ð»Ð¾Ð²Ð¸

url.address, url.domain

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

health_check_method, health_result, health_request_timeout, health_type_request,

health_answer_timeout, count_interval, max_event_count, scond_type

Ð Ñ Ð¸Ð¼ÐµÑ

```
[scenarios "Scenario Rules"]
% ---------------- 1 ----------------
def scenario_cond example_scenario_define
    category = (lib.category(Threats), "Advertisements & Pop-Ups") count_interval(10) max_event_co
    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) sc
    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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²Ð¾Ðº Ð½Ð°Ð³Ñ Ñ Ð·ÐºÐ¸ TCP/UDP

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð£Ñ Ð»Ð¾Ð²Ð¸Ñ

src.zone, src.geoip, src.ip, src.mac.

service, url.address, url.port.

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled.

scheduler, real_server, ipvs_fallback

monitor_kind, monitor_service, monitor_request, monitor_response, monitor_interval,

monitor_timeout, monitor_failurecount

Ð Ñ Ð¸Ð¼ÐµÑ

```
[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")
```

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²Ð¾Ðº ICAP

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, profile.

Ð Ñ Ð¸Ð¼ÐµÑ

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

Ð Ñ Ð°Ð²Ð¸Ð»Ð° Ð±Ð°Ð»Ð°Ð½Ñ Ð¸Ñ Ð¾Ð²ÐºÐ¸ reverse-Ð¿Ñ Ð¾ÐºÑ Ð¸

Ð Ñ ÐµÑ Ð¸ÐºÑ Ñ

Ð Ð¼Ñ	Ð Ð¿Ð¸Ñ Ð°Ð½Ð¸Ðµ
OK	Ð Ñ ÐµÐ³Ð´Ð° Ð Ð .

Ð¡Ð²Ð¾Ð¹Ñ Ñ Ð²Ð°

name, desc, enabled, profile.

Ð Ñ Ð¸Ð¼ÐµÑ

```
[reverseproxy_balancing "Reverse proxy load balancing Rules"]
% ----------------- 1 -----------------
OK \
    profile("Example reverse proxy server") \
    enabled(true) \
    name("Reverse-proxy load balancing")
```

Ð¡Ð¿Ð¸Ñ Ð¾Ðº Ð¿Ð¾Ð´Ð´ÐµÑ Ð¶Ð¸Ð²Ð°ÐµÐ¼Ñ Ñ HTTP-Ð·Ð°Ð³Ð¾Ð»Ð¾Ð²ÐºÐ¾Ð²

Version: 7.x
Product: NGFW

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			

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