??????? ? QoS

Connection tracking

????????

Отслеживание соединений (Connection tracking) позволяет ядру отслеживать все логические сетевые соединения или сессии и таким образом связывать все пакеты, которые могут составлять это соединение.

NAT полагается на эту информацию для однородного преобразования всех связанных пакетов.

Благодаря отслеживанию соединений можно использовать функциональность stateful firewall (состояния сеансов) даже с таким stateless протоколом, как UDP.

??????? ????????, ???????? ?? ???????????? ??????????:

- NAT
- firewall:
  - connection-bytes
  - connection-mark
  - connection-type
  - connection-state
  - connection-limit
  - connection-rate
  - layer7-protocol
  - new-connection-mark
  - tarpit

Список отслеживаемых соединений можно увидеть в `/ip firewall connection` для IPv4 и `/ipv6 firewall connection` для IPv6.

```
[admin@3C22-atombumba] /ip firewall connection> print
Flags: S - seen-reply, A - assured
# PR.. SRC-ADDRESS           DST-ADDRESS           TCP-STATE  TIMEOUT
0    udp 10.5.8.176:5678    255.255.255.255:5678  0s
1    udp 10.5.101.3:646     224.0.0.2:646         5s
2    ospf 10.5.101.161      224.0.0.5             9m58s
3    udp 10.5.8.140:5678    255.255.255.255:5678  8s
4    SA  tcp 10.5.101.147:48984 10.5.101.1:8291   established 4m59s

[admin@3C22-atombumba] /ipv6 firewall connection> print
Flags: S - seen reply, A - assured
# PRO.. SRC-ADDRESS                                DST-ADDRESS
0    udp fe80::d6ca:6dff:fe77:3698                ff02::1
1    udp fe80::d6ca:6dff:fe98:7c28                ff02::1
```

```
2    ospf fe80::d6ca:6dff:fe73:9822                ff02::5
```

????????? ??????????

На основе записей в таблице соединений входящий пакет может быть отнесён к одному из следующих состояний соединения: **new** (новое), **invalid** (недопустимое), **established** (установленное), **related** (связанное) или **untracked** (неотслеживаемое).

Существует два случая, когда пакет считается новым:

- в случае **бессостоя́тельных (stateless)** протоколов, например UDP — когда в таблице соединений нет записи для данного трафика;
- в случае **состоя́тельных (stateful)** протоколов, например TCP — новый пакет, открывающий соединение, всегда является TCP-пакетом с установленным флагом **SYN**.

Если пакет не новый, он может относиться к уже установленному (**established**) или связанному (**related**) соединению, либо не относиться ни к одному, что делает его **invalid**.

Пакет в состоянии **established** принадлежит существующему соединению, зарегистрированному в таблице connection tracking. Состояние **related** аналогично, но пакет связан с другим существующим соединением — например, это может быть ICMP-пакет с сообщением об ошибке или пакет для передачи данных FTP.

Состояние **untracked** — особый случай, когда с помощью правил таблицы **RAW** определён трафик, исключённый из отслеживания соединений (connection tracking). Такие правила позволяют пропускать определённые пакеты без обработки connection tracking, что повышает скорость обработки трафика устройством.

Любой другой пакет, не соответствующий указанным состояниям, считается **invalid** (недопустимым) и, как правило, должен быть отброшен.

Исходя из этого, можно создать базовый набор фильтрующих правил, который повысит производительность и снизит нагрузку на CPU: принимать пакеты состояний **established** и **related**, отбрасывать **invalid**, и применять детализированную фильтрацию только к **new**-пакетам.

```
ip firewall filter add chain=input connection-state=invalid action=drop comment="Drop Invalid co
add chain=input connection-state=established,related,untracked action=accept comment="Allow Esta
```

Такой набор правил не должен применяться на маршрутизаторах с асимметричной маршрутизацией, так как асимметрично маршрутизируемые пакеты могут быть признаны недействительными и отброшены.

FastTrack

IPv4 FastTrack — это специальный обработчик, который обходит стандартные возможности Linux, позволяя ускорить пересылку пакетов. Этот обработчик используется для соединений **TCP** и **UDP**, помеченных действием `fasttrack-connection`. Обработчик FastTrack поддерживает NAT (SNAT, DNAT или оба).

Следует отметить, что не все пакеты соединения могут быть обработаны FastTrack, поэтому вероятно, что некоторые пакеты будут идти по "медленному" пути даже при помеченном для FastTrack соединении. Это причина, по которой действие `fasttrack-connection` обычно сопровождается идентичным правилом с

```
action=accept
```

Пакеты, проходящие через **FastTrack**, обходят следующие механизмы системы:

- фаервол (firewall);
- отслеживание соединений (connection tracking);
- простые очереди (simple queues);
- дерево очередей (queue tree) с параметром `parent=global`;
- учёт трафика (IP accounting);
- шифрование IPSec;
- универсальный клиент Hotspot;
- назначение VRF.

Администратор должен самостоятельно убедиться, что использование **FastTrack** не конфликтует с другими элементами конфигурации.

??????????

Механизм **IPv4 FastTrack** активируется, если выполняются следующие условия:

- не используется конфигурация интерфейсов **mesh** или **metarouter**;
- не запущены инструменты **sniffer**, **torch** или **traffic generator**;
- инструмент **/tool mac-scan** не используется активно;
- инструмент **/tool ip-scan** не используется активно;
- в разделе **IP → Settings** включены опции **FastPath** и **Route Cache**.

??????

Например, для SOHO маршрутизаторов с заводской конфигурацией, вы можете применить FastTrack ко всему LAN-трафику одним правилом, размещенным в начале фильтра файрвола. Также требуется такое же правило с действием accept:

```
/ip firewall filter add chain=forward action=fasttrack-connection connection-state=established,r
/ip firewall filter add chain=forward action=accept connection-state=established,related
```

- Соединение обрабатывается FastTrack, пока оно не закрыто, не истекло по таймауту или не перезагружен маршрутизатор.

- Пустые правила исчезнут только после удаления/отключения правил FastTrack и перезагрузки маршрутизатора.
- Когда FastPath и FastTrack включены одновременно, активен может быть только один.
- Очереди (кроме Queue Trees, привязанных к интерфейсам), фильтр файрвола и правила mangle не применяются к FastTrack-трафику.

????????? ???????????? ??????????

Настройки отслеживания соединений управляются из меню `/ip firewall connection tracking`.

????????

Свойство	Описание
enabled (yes | no | auto; По умолчанию: auto)	Позволяет включать или отключать отслеживание соединений. Если отключено, перечисленные выше функции файрвола не работают. Если установлено в "auto", отслеживание соединений отключено до тех пор, пока не будет добавлено хотя бы одно правило файрвола.
liberal-tcp-tracking (yes | no; По умолчанию: no)	Включает или отключает либеральное отслеживание TCP соединений, переключая параметр ядра. Включение может позволить принять нарушенные пакеты, которые в противном случае считались бы ошибочными.
loose-tcp-tracking (yes; По умолчанию: yes)	
tcp-syn-sent-timeout (time; По умолчанию: 5s)	Таймаут SYN TCP.
tcp-syn-received-timeout (time; По умолчанию: 5s)	Таймаут SYN TCP.
tcp-established-timeout (time; По умолчанию: 1d)	Время, после которого считается таймаут установленного TCP соединения.
tcp-fin-wait-timeout (time; По умолчанию: 10s)	
tcp-close-wait-timeout (time; По умолчанию: 10s)	
tcp-last-ack-timeout (time; По умолчанию: 10s)	
tcp-time-wait-timeout (time; По умолчанию: 10s)	
tcp-close-timeout (time; По умолчанию: 10s)	
udp-timeout (time; По умолчанию: 30s)	Задержка для UDP соединений, которые видели пакеты только в одном направлении.
udp-stream-timeout (time; По умолчанию: 3m)	Задержка для UDP соединений, которые видели пакеты в обоих направлениях.
icmp-timeout (time; По умолчанию: 10s)	Таймаут для ICMP соединений.
generic-timeout (time; По умолчанию: 10m)	Таймаут для всех остальных соединений.

?????? ??? ??????

Свойство	Описание
max-entries (integer)	Максимальное количество записей, которые может содержать таблица отслеживания соединений. Это значение зависит от объема установленной оперативной памяти. Система не создает таблицу максимального размера при запуске, но может ее увеличить при необходимости, если есть свободная память, не превышая 1048576 записей.
total-entries (integer)	Текущее количество соединений в таблице отслеживания.

?????? ??????????

Список отслеживаемых соединений можно увидеть в `/ip firewall connection` для IPv4 и `/ipv6 firewall connection` для IPv6.

????????

Все свойства в списке соединений доступны только для чтения.

Свойство	Описание
assured (yes | no)	Указывает, что соединение гарантировано и не будет удалено при достижении максимального количества отслеживаемых соединений.
confirmed (yes | no)	Соединение подтверждено и пакет был отправлен с устройства.
connection-mark (string)	Отметка соединения, установленная правилом mangle.
connection-type (pptp | ftp)	Тип соединения; поле пустое, если невозможно определить тип соединения.
dst-address (ip)	Адрес назначения.
dst-port (integer)	Порт назначения.
dstnat (yes | no)	Соединение прошло через DST-NAT (например, перенаправление портов).
dying (yes | no)	Соединение в процессе окончания из-за таймаута.
expected (yes | no)	Соединение установлено с помощью помощников соединений (предопределенных правил сервиса).
fasttrack (yes | no)	Соединение обработано с помощью FastTrack.
gre-key (integer)	Данные поля GRE Key.
gre-protocol (string)	Протокол инкапсулированного полезного груза.
gre-version (string)	Версия протокола GRE, используемая в соединении.
hw-offload (yes | no)	Аппаратно ускоренное соединение.
icmp-code (string)	Поле ICMP Code.

Свойство	Описание
icmp-id (integer)	ID ICMP.
icmp-type (integer)	Номер типа ICMP.
orig-bytes (integer)	Количество байт, отправленных с исходного адреса по этому соединению.
orig-fasttrack-bytes (integer)	Количество байт FastTrack, отправленных с исходного адреса.
orig-fasttrack-packets (integer)	Количество пакетов FastTrack, отправленных с исходного адреса.
orig-packets (integer)	Количество пакетов, отправленных с исходного адреса по соединению.
orig-rate (integer)	Скорость передачи данных от исходного адреса.
protocol (string)	Тип IP протокола.
repl-bytes (integer)	Количество байт, полученных от адреса назначения.
repl-fasttrack-bytes (string)	Количество байт FastTrack, полученных от адреса назначения.
repl-fasttrack-packets (integer)	Количество пакетов FastTrack, полученных от адреса назначения.
repl-packets (integer)	Количество пакетов, полученных от адреса назначения.
repl-rate (string)	Скорость приема данных от адреса назначения.
reply-dst-address (ip)	Адрес назначения, ожидаемый для обратных пакетов.
reply-dst-port (integer)	Порт назначения, ожидаемый для обратных пакетов.
reply-src-address (ip)	Исходный адрес, ожидаемый для обратных пакетов.
reply-src-port (integer)	Исходный порт, ожидаемый для обратных пакетов.
seen-reply (yes | no)	Адрес назначения ответил на исходящий адрес.
src-address (ip)	Исходный адрес.
src-port (integer)	Исходный порт.
srcnat (yes | no)	Соединение проходит через SRC-NAT, включая пакеты, замаскированные через NAT.
tcp-state (string)	Текущее состояние TCP соединения.
timeout (time)	Время до удаления соединения из списка.
uses-helper (yes | no)	Применен помощник из "IP/Firewall/Service Port" к этому соединению.

Firewall

?????? Firewall

Фаервол в MikroTik реализует как stateful (с отслеживанием состояния) (с использованием connection tracking), так и stateless (без отслеживания состояния) фильтрацию пакетов, обеспечивая функции безопасности для управления потоком данных к маршрутизатору, от него и через него. Вместе с технологией Network Address Translation (NAT) фаервол служит инструментом для предотвращения несанкционированного доступа к напрямую подключённым сетям и самому маршрутизатору, а также фильтром для исходящего трафика.

Сетевые фаерволы защищают внутренние сети от внешних угроз. Когда объединяются разные сети, всегда существует риск, что кто-то извне проникнет в локальную сеть (LAN). Такие вторжения могут привести к краже и распространению приватных данных, изменению или уничтожению ценной информации, либо полному удалению данных с дисков. Фаервол используется для предотвращения или минимизации рисков безопасности, связанных с подключением к другим сетям. Правильно настроенный фаервол играет ключевую роль в создании эффективной и безопасной сетевой инфраструктуры.

MikroTik RouterOS обладает очень мощной реализацией фаервола с возможностями, включая:

- проверку пакетов без отслеживания состояния (stateless packet inspection);
- проверку пакетов с отслеживанием состояния (stateful packet inspection);
- обнаружение протоколов уровня 7 (Layer-7 protocol detection);
- фильтрацию пиринговых протоколов (peer-to-peer protocols filtering);
- классификацию трафика по:
    - MAC-адресу источника;
    - IP-адресам (сети или списки) и типам адресов (broadcast, local, multicast, unicast);
    - порту или диапазону портов;
    - IP-протоколам;
    - опциям протокола (ICMP type и code, TCP-флаги, IP options и MSS);
    - интерфейсу, через который пакет пришёл или ушёл;
    - внутренним отметкам потока и соединения (flow and connection marks);
    - DSCP байту;
    - содержимому пакета;
    - скорости поступления пакетов и номерам последовательности;
    - размеру пакета;
    - времени поступления пакета;
    - и многое другое!

Межсетевой экран разделен на три основных модуля:

- filter/raw — используется для отказа в трафике на основе настроенных политик. Фильтрация в RAW таблицах позволяет экономить ресурсы, если не требуется отслеживание соединений.
- mangle — используется для маркировки определенных соединений, пакетов, потоков, установления приоритетов и выполнения других задач.
- nat — используется для установки правил преобразования адресов, перенаправлений и проброса портов.

???????

Правила фильтрации межсетевого экрана сгруппированы в цепочки. Это позволяет сопоставить пакет с одним общим критерием в одной цепочке, а затем передать на обработку по другим критериям в другую цепочку.

Например, пакет должен быть сопоставлен с парой IP адрес:порт. Конечно, это можно достичь, добавив столько правил с совпадением по IP адресу и порту, сколько требуется, в цепочку forward, но лучшим способом может быть добавление одного правила, которое соответствует трафику от определенного IP адреса. Затем можно добавить правила, которые проводят сопоставление по отдельным портам, в цепочку mychain без указания IP адресов.

```
/ip firewall filter add chain=mychain protocol=tcp dst-port=22 action=accept
add chain=mychain protocol=tcp dst-port=23 action=accept
add chain=input src-address=1.1.1.2/32 jump-target="mychain"
```

При обработке цепочки правила берутся из цепочки в порядке их перечисления сверху вниз. Если пакет соответствует критериям правила, то выполняется указанное действие, и остальные правила в этой цепочке не обрабатываются (исключением является действие passthrough и некоторые операции Mangle).

Если пакет не соответствует ни одному правилу в цепочке, то он принимается.

Каждый модуль межсетевого экрана имеет свои предопределенные цепочки:

- **raw**:
  - prerouting
  - output
- **filter**:
  - input
  - forward
  - output
- **mangle**:
  - prerouting

    - input
    - forward
    - output
    - postrouting

- **nat**:
    - srcnat
    - dstnat

Более детальная обработка пакетов в RouterOS описана в диаграмме Packet Flow в RouterOS.

Firewall

Filter

????????

Раздел Filter межсетевого экрана используются для разрешения или блокировки отдельных пакетов, перенаправленных в вашу локальную сеть, исходящих от вашего маршрутизатора или направленных к нему.

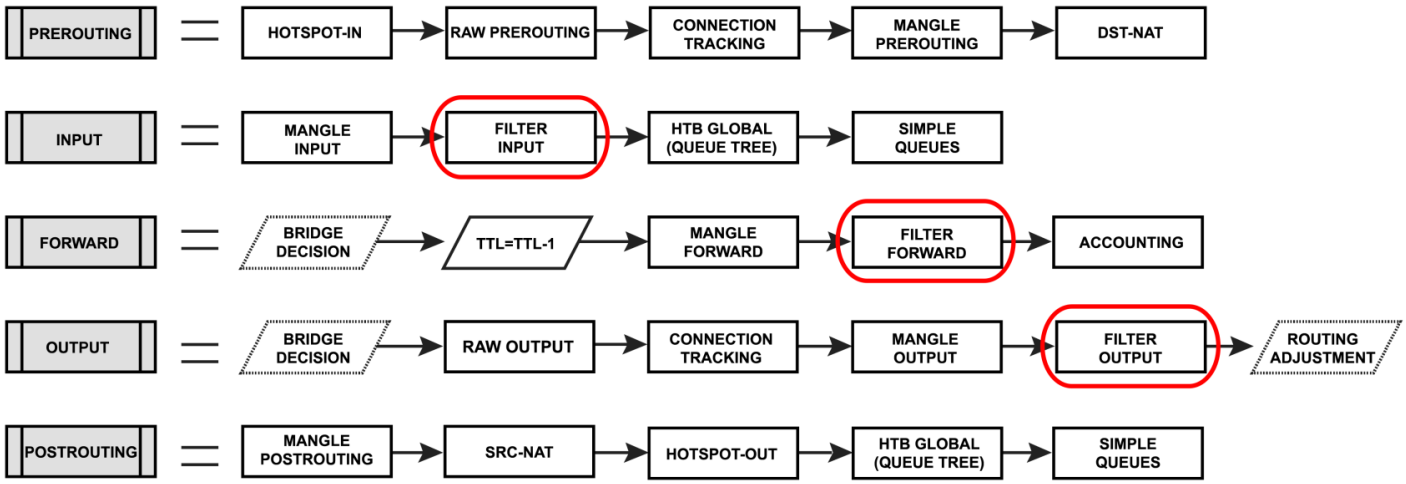
Существует два метода настройки фильтрации:

- разрешать конкретный трафик и блокировать всё остальное;
- блокировать только вредоносный трафик, разрешая всё остальное.

Оба метода имеют свои плюсы и минусы, например, с точки зрения безопасности первый метод гораздо безопаснее, но требует вмешательства администратора при необходимости разрешения трафика для новой службы. Эта стратегия даёт хороший контроль над трафиком и снижает вероятность нарушения безопасности из-за ошибочной настройки службы.

С другой стороны, при защите клиентской сети было бы административным кошмаром разрешать все возможные службы, используемые пользователями. Поэтому тщательное планирование межсетевого экрана необходимо для сложных настроек.

Фильтр межсетевого экрана состоит из трёх предопределённых цепочек, которые нельзя удалять:



- **input** — используется для обработки пакетов, проходящих через один из интерфейсов с адресом назначения, который является одним из адресов маршрутизатора. Пакеты, проходящие через маршрутизатор, не обрабатываются правилами цепочки input (*входящие на маршрутизатор*).

- **forward** — используется для обработки пакетов, проходящих через маршрутизатор.
- **output** — используется для обработки пакетов, исходящих из маршрутизатора через один из интерфейсов. Пакеты, проходящие через маршрутизатор, не обрабатываются правилами цепочки output (*исходящие из маршрутизатора*).

Конфигурация фильтра межсетевого экрана доступна из меню `ip/firewall/filter` для IPv4 и `ipv6/firewall/filter` для IPv6.

?????? ???????????? ???????????? ??????

Рассмотрим базовый пример межсетевого экрана для защиты самого маршрутизатора и клиентов за ним, как для протоколов IPv4, так и IPv6.

IPv4 ?????????? ?????

?????? ?????? ?????????????

Основные правила настройки межсетевого экрана:

- работать с `new` соединениями, чтобы снизить нагрузку на маршрутизатор;
- разрешать то, что нужно, и блокировать всё остальное (`drop`);
- `log=yes` можно установить для ведения журнала некоторых атак, но это может увеличить нагрузку на CPU при сильных атаках.

Всегда начинаем с разрешения уже установленных и связанных соединений, поэтому первое правило должно принимать "established" и "related" соединения:

```
/ip firewall filter add action=accept chain=input comment="default configuration" connection-state=established,related
```

Затем разрешаем некоторые новые соединения. В примере разрешается доступ по протоколу ICMP с любого адреса и всё остальное только из диапазона адресов 192.168.88.2-192.168.88.254. Для этого создаётся список адресов и два правила фильтра межсетевого экрана:

```
/ip firewall address-list add address=192.168.88.2-192.168.88.254 list=allowed_to_router
/ip firewall filter add action=accept chain=input src-address-list=allowed_to_router
/ip firewall filter add action=accept chain=input protocol=icmp
```

И, наконец, блокируем всё остальное:

```
add action=drop chain=input
```

Полный набор только что созданных правил:

```
/ip firewall filter add action=accept chain=input comment="default configuration" connection-
state=established,related
add action=accept chain=input src-address-list=allowed_to_router
add action=accept chain=input protocol=icmp
add action=drop chain=input

/ip firewall address-list add address=192.168.88.2-192.168.88.254 list=allowed_to_router
```

?????? ????????? ????????? ???? (LAN)

Подход к защите пользователей схож, за исключением того, что в этом случае мы блокируем нежелательный трафик и разрешаем всё остальное.

Сначала создаём `address-list` с именем "not_in_internet", который будет использоваться в правилах фильтра межсетевого экрана:

```
/ip firewall address-list add address=0.0.0.0/8 comment=RFC6890 list=not_in_internet
add address=172.16.0.0/12 comment=RFC6890 list=not_in_internet
add address=192.168.0.0/16 comment=RFC6890 list=not_in_internet
add address=10.0.0.0/8 comment=RFC6890 list=not_in_internet
add address=169.254.0.0/16 comment=RFC6890 list=not_in_internet
add address=127.0.0.0/8 comment=RFC6890 list=not_in_internet
add address=224.0.0.0/4 comment=Multicast list=not_in_internet
add address=198.18.0.0/15 comment=RFC6890 list=not_in_internet
add address=192.0.0.0/24 comment=RFC6890 list=not_in_internet
add address=192.0.2.0/24 comment=RFC6890 list=not_in_internet
add address=198.51.100.0/24 comment=RFC6890 list=not_in_internet
add address=203.0.113.0/24 comment=RFC6890 list=not_in_internet
add address=100.64.0.0/10 comment=RFC6890 list=not_in_internet
add address=240.0.0.0/4 comment=RFC6890 list=not_in_internet
add address=192.88.99.0/24 comment="6to4 relay Anycast [RFC 3068]" list=not_in_internet
```

Краткое объяснение правил фильтра межсетевого экрана:

- пакеты с `connection-state=established,related` добавляются в FastTrack для ускорения передачи данных, фильтр будет работать только с новыми соединениями;
- блокировать `invalid` соединения и вести журнал с префиксом "invalid";
- блокировать попытки доступа к непубличным адресам из локальной сети с применением `address-list=not_in_internet`, "bridge" — интерфейс локальной сети, логировать при попытках с префиксом "!public_from_LAN";
- блокировать входящие пакеты, которые не проходят NAT, интерфейс ether1 — публичный, логировать с префиксом "!NAT";
- прыжок в цепочку ICMP для отброса нежелательных ICMP сообщений;

- блокировать входящие с Интернета непубличные IP адреса, интерфейс ether1 — публичный, логировать попытки с префиксом "!public";
- блокировать пакеты из локальной сети, которые не имеют IP в подсети LAN 192.168.88.0/24.

```
/ip firewall filter add action=fasttrack-connection chain=forward comment=FastTrack
connection-state=established,related
add action=accept chain=forward comment="Established, Related" connection-
state=established,related
add action=drop chain=forward comment="Drop invalid" connection-state=invalid log=yes log-
prefix=invalid
add action=drop chain=forward comment="Drop tries to reach not public addresses from LAN" dst-
address-list=not_in_internet in-interface=bridge log=yes log-prefix=!public_from_LAN out-
interface=!bridge
add action=drop chain=forward comment="Drop incoming packets that are not NAT`ted" connection-
nat-state=!dstnat connection-state=new in-interface=ether1 log=yes log-prefix=!NAT
add action=jump chain=forward protocol=icmp jump-target=icmp comment="jump to ICMP filters"
add action=drop chain=forward comment="Drop incoming from internet which is not public IP" in-
interface=ether1 log=yes log-prefix=!public src-address-list=not_in_internet
add action=drop chain=forward comment="Drop packets from LAN that do not have LAN IP" in-
interface=bridge log=yes log-prefix=LAN_!LAN src-address=!192.168.88.0/24
```

Разрешаем только нужные коды ICMP в цепочке "icmp":

```
/ip firewall filter add chain=icmp protocol=icmp icmp-options=0:0 action=accept comment="echo
reply"
add chain=icmp protocol=icmp icmp-options=3:0 action=accept comment="net unreachable"
add chain=icmp protocol=icmp icmp-options=3:1 action=accept comment="host unreachable"
add chain=icmp protocol=icmp icmp-options=3:4 action=accept comment="host unreachable
fragmentation required"
add chain=icmp protocol=icmp icmp-options=8:0 action=accept comment="allow echo request"
add chain=icmp protocol=icmp icmp-options=11:0 action=accept comment="allow time exceed"
add chain=icmp protocol=icmp icmp-options=12:0 action=accept comment="allow parameter bad"
add chain=icmp action=drop comment="deny all other types"
```

IPv6 ?????????? ?????

?????? ?????? ?????????????

Очень похожа на настройку IPv4, за исключением того, что для корректной работы IPv6 требуется поддержка большего числа протоколов.

Сначала создаём `address-list`, из которого разрешаем доступ к устройству:

```
/ipv6 firewall address-list add address=fd12:672e:6f65:8899::/64 list=allowed
```

Краткое объяснение правил фильтра IPv6:

- работать с `new` пакетами, разрешать `established/related` пакеты;
- блокировать `link-local` адреса с публичного интернет-интерфейса или списка интерфейсов;
- разрешать доступ к маршрутизатору с `link-local` адресов, разрешать `multicast` адреса для целей управления, разрешать ваш исходный `address-list` для доступа к маршрутизатору;
- блокировать всё остальное.

```
/ipv6 firewall filter add action=accept chain=input comment="allow established and related"
connection-state=established,related
add chain=input action=accept protocol=icmpv6 comment="accept ICMPv6"
add chain=input action=accept protocol=udp port=33434-33534 comment="defconf: accept UDP
traceroute"
add chain=input action=accept protocol=udp dst-port=546 src-address=fe80::/10 comment="accept
DHCPv6-Client prefix delegation."
add action=drop chain=input in-interface=in_interface_name log=yes log-
prefix=dropLL_from_public src-address=fe80::/10
add action=accept chain=input comment="allow allowed addresses" src-address-list=allowed
add action=drop chain=input
/ipv6 firewall address-list add address=fe80::/16 list=allowed
add address=xxxx::/48 list=allowed
add address=ff02::/16 comment=multicast list=allowed
```

В некоторых настройках, где используется DHCPv6 ретранслятор, src-адрес пакетов может не принадлежать диапазону link-local. В таком случае параметр src-address правила №4 должен быть удалён или изменён для разрешения адреса ретранслятора.

?????? ????????? ????????? ???? (LAN)

Этот шаг важнее, чем в IPv4. В IPv4 клиенты обычно имеют адреса из локального диапазона и работают через NAT с публичным IP, поэтому они недоступны напрямую из публичных сетей.

В IPv6 всё иначе. В большинстве распространённых настроек включенный IPv6 делает клиентов доступными из публичных сетей, поэтому обязательны правила фильтра межсетевого экрана для защиты клиентов.

Вкратце, базовая защита LAN должна:

- принимать `established/related` и работать с `new` пакетами;
- блокировать `invalid` пакеты;
- принимать ICMPv6 пакеты;
- разрешать новые соединения, исходящие только от клиентов в публичную сеть;
- блокировать всё остальное.

```
/ipv6 firewall filter add action=accept chain=forward comment=established,related connection-state=established,related
add action=drop chain=forward comment=invalid connection-state=invalid log=yes log-prefix=ipv6,invalid
add action=accept chain=forward comment=icmpv6 in-interface=!in_interface_name protocol=icmpv6
add action=accept chain=forward comment="local network" in-interface=!in_interface_name src-address-list=allowed
add action=drop chain=forward log-prefix=IPV6
```

????????????? (Matchers)

Все свойства сопоставителей общие и перечислены здесь.

???????? (Actions)

Таблицы ниже показывают список специфичных для фильтра действий и связанных свойств. Другие действия перечислены здесь.

Свойство	Описание
action (имя действия; по умолчанию: accept)	
hw-offload (нет | да; по умолчанию: да)	Включает или выключает аппаратное ускорение FastTrack. Поддерживается только на коммутаторах с ускорением FastTrack и когда установлен `action=fasttrack-connection`.
reject-with (icmp-no-route | icmp-admin-prohibited | icmp-not-neighbour | icmp-address-unreachable | icmp-port-unreachable | tcp-reset | icmp-err-src-routing-header | icmp-headers-too-long ; по умолчанию: icmp-no-route)	Определяет ICMP-ошибку, которая будет отправлена назад, если пакет отклонён. Применяется если ...

?????????? RAW

RAW таблица межсетевого экрана позволяет выборочно пропускать или блокировать пакеты до отслеживания соединений, значительно снижая нагрузку на CPU. Это очень полезно для смягчения DoS/DDoS атак.

Конфигурация RAW фильтра доступна из меню `ip/firewall/raw` для IPv4 и `ipv6/firewall/raw` для IPv6.

RAW таблица не имеет сопоставителей, зависящих от отслеживания соединений (например, connection-state, layer7 и др.).

Если пакет помечен для обхода отслеживания соединений, пакет не фрагментируется.

RAW таблица может иметь правила только в двух цепочках:

- **prerouting** — используется для обработки любого пакета, входящего в маршрутизатор;
- **output** — используется для обработки пакетов, исходящих из маршрутизатора через любой интерфейс. Пакеты, проходящие через маршрутизатор, не обрабатываются правилами цепочки output.

Имеет одно специфическое действие:

Свойство	Описание
action (название действия; по умолчанию: accept)	notrack — не отправлять пакет в систему отслеживания соединений (connection tracking). Полезно, когда необходимо использовать обычный фаервол, но отслеживание соединений не требуется.

?????? ??????? ????????? RAW

Предположим, у нас есть настройка OSPF, но из-за отслеживания соединений у OSPF возникают проблемы с соседством. Можно использовать правила RAW для исправления, не отправляя OSPF пакеты в отслеживание соединений:

```
/ip firewall raw add chain=prerouting protocol=ospf action=notrack
add chain=output protocol=ospf action=notrack
```