Building Your First Firewall

- Overview
  - Ipv4 firewall
    - Protect the router itself
    - Protect the LAN devices
  - IPv6 firewall
    - Protect the router itself
    - Protect the LAN devices

Overview

We strongly suggest keeping the default firewall on. Here are a few adjustments to make it more secure. Make sure you configure additional changes when you completely understand the benefit of these particular firewall rules.

To see the default firewall rules through the CLI you can type:

```
/system default-configuration print
```

Ipv4 firewall

Protect the router itself

- work with *new* connections to decrease load on a router;
- create *address-list* for IP addresses, that are allowed to access your router;
- enable ICMP access (optionally);
- drop everything else, *log=yes* might be added to log packets that hit the specific rule;

```
/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
```

Protect the LAN devices

We will create *address-list* with name "not_in_internet" which we will use for the firewall filter rules:

```
/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
```

Brief firewall filter rule explanation:

- packets with *connection-state=established,related* added to FastTrack for faster data throughput, firewall will work with new connections only;
- drop *invalid* connection and log them with prefix "invalid";
- drop attempts to reach not public addresses from your local network, apply *address-list=not_in_internet* before, "bridge" is local network interface, log=yes attempts with prefix "!public_from_LAN";
- drop incoming packets that are not NAT`ed, ether1 is public interface, log attempts with "!NAT" prefix;
- jump to ICMP chain to drop unwanted ICMP messages
- drop incoming packets from the Internet, which are not public IP addresses, ether1 is a public interface, log attempts with prefix "!public";
- drop packets from LAN that does not have LAN IP, 192.168.88.0/24 is local network used subnet;

```
/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
```

Allow only needed icmp codes in "icmp" chain:

```
/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 firewall

Protect the router itself

Create an address-list from which you allow access to the device:

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

Brief IPv6 firewall filter rule explanation:

- work with *new* packets, accept *established/related* packets;
- drop *link-local* addresses from Internet(public) interface/interface-list;
- accept access to a router from *link-local* addresses, accept *multicast* addresses for management purposes, accept your source *address-list* for router access;
- drop anything else;

```
/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::/16 comment="accept DHCPv6-Client
prefix delegation."
add action=drop chain=input in-interface=sit1 log=yes log-prefix=dropLL_from_public src-address=fe80::/16
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
```

Protect the LAN devices

Enabled IPv6 puts your clients available for public networks, set proper firewall to protect your customers.

- accept *established/related* and work with *new* packets;
- drop *invalid* packets and put prefix for rules;
- accept ICMP packets;
- accept *new* connection from your clients to the Internet;
- drop everything else.

```
/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=!sit1 protocol=icmpv6
add action=accept chain=forward comment="local network" in-interface=!sit1 src-address-list=allowed
add action=drop chain=forward log-prefix=IPV6
```