

# ipfw command (IP Firewall)



Successfully pass the free certification exam at IW Academy and become an Infinet Certified Engineer.

[To the certification exam](#)

- [Description](#)
- [Parameters](#)
- [Examples](#)

## Description

IP Firewall is a mechanism of filtering packets crossing an IP network node, according to different criteria. System administrator may define a set of incoming filters (addincoming) and a set of outgoing filters (addoutgoing). The incoming filters determine which packets may be accepted by the node. The outgoing filters determine which packets may be forwarded by the node as a result of routing. Each filter describes a class of packets and defines how these packets should be processed (reject and log, accept, accept and log).

Packets can be filtered based on the following criteria:

- Protocol (IP, TCP, UDP, ICMP, ARP);
- Source address and/or destination address (and port numbers for TCP and UDP);
- The network interface it arrived on;
- Whether the packet is a TCP/IP connection request (a packet attempting to initiate a TCP/IP session) or not;
- Whether the packet is a head, tail or intermediate IP fragment;
- Whether the packet has certain IP options defined or not;
- The MAC address of the destination station or of the source station.

### Syntax:

```
list
show | reset
rearrange [STEP]
flush
quiet | -quiet
del RULE_NUMBER
dump RULE_NUMBER
mov RULE_A RULE_B
add[out] [NUM] [IFNAME] [chN] rules...


rules: [{setpri|addpri}=[N]] accept|reject|rpfilter|pass [log]
      [vlan={N|any|$ACL}] [dot1p=N] [swg=N] [ether={X|any}] [dscp=N|tos=N] [prf]
      -f "pcap filter expression"
      |
      PROTO from [not] ADDR [PORTs] to [not] ADDR [PORTs]

PROTO: [all] | tcp | udp | icmp | arp | proto NUMBER
ADDR: IP | $LOCAL | $ROUTE | $ACL | mac x:x:x:x:x }
PORTS: NUM[:NUM] [NUM] ...
```

## Parameters

Parameter	Description
<b>list</b>	Displays currently defined filters.
<b>show / reset</b>	Shows/resets filters statistic.
<b>rearrange [STEP]</b>	Renumbers all the filter rules with the increment "STEP" (default is 5).
<b>flush</b>	All currently defined filters in both the incoming and outgoing filter sets are removed. Filtering is disabled.

<b>quiet   -quiet</b>	Disables logging of rejected packets. Registration is enabled by default, to re-enable registration use " <i>ipfw -quiet</i> " command.
<b>del RULE_NUMBER</b>	Removes a filter from the appropriate list. The filter to be removed is specified by its number which can be seen using the " <i>ipfw list</i> " command.
<b>dump RULE_NUMBER</b>	Displays the compiled filter pseudocode specified in the PCAP format. Allows to assess visually the complexity / optimality, or the correctness of the resulting filter.
<b>mov RULE_A RULE_B</b>	Moves the filter with "A" number to position "B".
<b>add[out] [NUM] [IFNAME] [chN] rules...</b>	<p>Adds a filter to the incoming and outgoing (if "out" prefix appears) filter sets.</p> <ul style="list-style-type: none"> <li>"NUM" – specifies the new filter number in the list.</li> <li>"IFNAME" – network interface name, through which the packet enters the system.</li> <li>"chN" – in case of matching this argument, the service channel N created by the "<i>qm chN</i>" command will be assigned to the packet.</li> </ul>
<b>rules: [[setpri addpri]= [N]] accept reject rpfilter pass [log] [vlan= {N any \$ACL}] [dot1p=N] [swg=N] [ether= {X any}] [dscp=N tos=N] [prf] -f "pcap filter expression"</b>	<ul style="list-style-type: none"> <li>"setpri=[N]" – sets a packet priority to the value specified in the argument.</li> <li>"addpri=[N]" – changes a priority only in case it is higher than the previous one. The "addpri" parameter allows only to increase priority.</li> <li>"accept" – the packet is processed by the system, in spite of other IP Firewall rules.</li> <li>"reject" – the packet is discarded.</li> <li>"rpfilter" – ensures that the sender of the package is accessible via the interface through which package it received in the system. If the filter fails, the packet processing continues, if not fails the packet is destroyed. This filter can be inserted into the list of rules first.</li> <li>"pass" – allows a packet to pass a rule executing the related actions of this rule and continue with other rules in the list.</li> <li>"log" – adds filter action records in the system log.</li> <li>"vlan=" – classifier that allows to analyze VLAN ID (allowable values 0-4095): <ul style="list-style-type: none"> <li>"N" – filter will skip tagged packets with the specified tag N.</li> <li>"any" – filter will skip all tagged packets with any VLAN ID.</li> <li>"\$ACL" – filter will skip tagged packets with VLAN tags, specified in "\$ACL" list (for more information see <a href="#">ACL command</a> description).</li> </ul> </li> <li>"dot1p=N" – allows to analyze 802.1p priority (allowable values 0-7).</li> <li>"swg=N" – allows to analyze switching group number.</li> <li>"ether={X any}" – allows to analyze packet type. If option "any" is selected, the filter accepts packets of all type.</li> <li>"dscp=N" – allows to analyze DSCP tag (allowable values 0-63).</li> <li>"tos=N" – allows to analyze TOS tag.</li> <li>"prf" – enables PRF interface traffic filtration.</li> <li>"-f "pcap filter expression" – allows to use pcap filters.</li> </ul>
<b>PROTO from [not] ADDR [PORTs] to [not] ADDR [PORTs]</b>	<p>Classifiers which defines direction of transmission from and/or to ID:</p> <ul style="list-style-type: none"> <li>"from" – source IP address.</li> <li>"to" – destination IP address.</li> <li>"not" – negative prefix, can be used after "from" and "to" keywords, its action will be applied to appropriate addresses only, but not to ports, if they are used in the command.</li> <li>"ADDR" – source or destination (endpoint) IP address. This field syntax depends on the "proto" value. If "proto" is "all" or "icmp", then "ADDR" contains address information. If "proto" is "udp" or "tcp", then "ADDR" contains address information and an optional ports list. Address information is specified as an IP address and optional mask. IP address should be set in a traditional numeric format (nn.nn.nn.nn). An optional mask can be set either as mask length in bits or as a numeric value in nnn.nnn.nnn.nnn format.</li> </ul> <p>Possible formats are:</p> <pre>nn . nn . nn . nn nn . nn . nn . nn : xxx . xxx . xxx . xxx nn . nn . nn . nn / NN</pre> <p>The "0/0" value specifies all possible IP addresses.</p>
<b>PROTO: [all]   tcp   udp   icmp   arp   proto NUMBER</b>	Sets some particular IP protocol, which is used for the filter. Possible values: TCP, UDP, ICMP, ARP or a numeric protocol value. ARP packets will always be passed for those IP addresses and ranges of IP addresses that are specified in accept filters, even if these filters are created for other packets types.

<b>ADDR: IP / \$LOCAL / \$ROUTE / \$ACL / mac x:x:x: x:x:x }</b>	<p>If it is necessary to create a filter which is applied to several network addresses or groups, it is more convenient to group all those addresses in one corresponding access list and specify the list name as an IP address ("<i>\$ACLRULE</i>"). Here are several predefined dynamic ACL lists:</p> <ul style="list-style-type: none"> <li>• "<i>\$LOCAL</i>" – includes all local addresses owned by the router. This list can be used for a convenient filter description which allow (or restrict) the access to the device.</li> <li>• "<i>\$ROUTE</i>" – contains system routes table excluding default route. When an IP address matches this list it means that this address has some specific route and default route will not be used in this case.</li> <li>• "<i>\$ACL</i>" – IP addresses or networks list this rule will be applied to.</li> </ul> <p>"<i>mac x:x:x:x:x:x</i>" – for the interfaces which have physical MAC addresses in Ethernet standard, it is possible to use a MAC address value with a key word "<i>mac</i>". For the incoming filters you can set only the source MAC address, and for outgoing – only the destination MAC address. Moreover, instead of a numeric value a key word "<i>\$BS</i>" can be used. In this case the corresponding BS MAC address (on which the CPE is configured) will be used.</p> <div style="border: 1px solid #f9e79f; padding: 10px; margin-top: 10px;"> <p> <b>NOTE</b></p> <p>Keep in mind that the rules which use MAC addresses for the incoming packets will be applied in the first turn, and the rules for the outgoing packets will be applied in the last turn.</p> </div>
<b>PORTS: NUM[: NUM] [NUM] ...</b>	<p>Is used to filter traffic by port numbers. Ports list allows to specify multiple ports in one command. The first element of the ports list can specify a range of numbers from smaller to greater, separated by a colon.</p>

Filters can be extended by using pcap-expressions.

#### Packet filtering rules

Each packet entering a router passes through a input filters set (or blocking filters). Packets accepted by the input filter set are further processed by the IP layer of the router kernel. If the IP layer determines that the packet should go further and not landing here, it hands the packet to the outgoing filters set (or forwarding filters).

Information on packets rejected by any filter is displayed on the operator's terminal, and the packets themselves are discarded without any notice to their sender.

A packet, "*advancing through*" a filters set is checked by every filter in the set, from the first one till the end of the set, or until the first matching filter. The algorithm is:

1. If the filter set is empty, the packet is accepted.
2. Otherwise, the first matching filter will make a decision about the packet. If it is an accept filter, the packet is accepted. If it's a reject filter, the packet is rejected (discarded).
3. If no filter has been found that matches the packet, it is accepted.

## Examples

Set filter which rejects all packets from the "1.1.1.1" IP address to the "2.2.2.2".

```
ipfw add reject all from 1.1.1.1 to 2.2.2.2
```

Set filter which rejects all packets from the "1.1.1.0/24" network to the "2.2.2.2" IP address.

```
ipfw add reject all from 1.1.1.0/24 to 2.2.2.2
```

Or

```
ipfw add reject all from 1.1.1.1:255.255.255.0 to 2.2.2.2
```

Set filter which rejects all packets from the "1.1.1.0" C class network, to any address (if pass through this router).

```
ipfw add reject all from 1.1.1.0/24 to 0/0
```

Set filter which allows all TCP packets to address an smtp service (mail agent) at the host with IP address "192.5.42.1". Port 25 corresponds to the SMTP service.

```
ipfw add accept tcp from 0/0 to 192.5.42.1 25
```

Set filter which accepts all TCP packets sended on the "1.1.1.1" IP address, if the destination port number is within the 900 to 5000 range or is equal to 25 (smtp) or 113 (ident).

```
ipfw add accept tcp from 0/0 to 1.1.1.1 900:5000 25 113
```

In the previous examples, the source address was used as main and the only criteria for the address reliability checking. Unfortunately, there is a possibility to send the packets from an unreliable address, substituting the return address with that you rely on (this attack method is called IP spoofing). It is clear that the checking only of the source address is not enough. It is necessary to check the packet path or, which is more practical, to check the interface through which the packet was accepted.

All inner network subnets, including a host address innerhost, are owned by the one network (or a network group). Let's imagine that outer network has no hosts which are within the range set up for the inner network. Therefore, all the packets that are accepted via rf5.0 interface of the router with firewall run on it and have the source address which is in the inner network addresses range must be blocked. The following command can perform this action, this filter will be applied only to those packets which come through rf5.0 interface. Packets which come through any other interface will not be blocked.

```
ipfw add rf5.0 reject all from innerhost/16 to 0/0
```

Additionally it is possible to block all packets with source address from the loopback network "127.0.0.0".

```
ipfw add rf5.0 reject all from 127.0.0.0/8 to 0/0
```

TCP/IP clients normally use port numbers between 900 and 5000 inclusive, leaving port numbers below 900 and above 5000 for servers. The following filters pair will bar access to your servers for any outside clients (assuming that all communications between your network and the external world pass through the rf5.0 interface).

Set the filter which accepts packets from external sources to ports from 900 to 5000 on the inner network hosts (normally assigned to internal clients). The second filter rejects all the rest.

```
ipfw add rf5.0 accept tcp from 0/0 to 0/0 900:5000  
ipfw add rf5.0 reject tcp from 0/0 to 0/0
```

## Title

Unlike the connection-oriented TCP protocol, the UDP protocol sends separate packets (datagrams). In this protocol every packet is transmitted independently from all others, and if there is a logical connection or session between a client and a server communicating through UDP, such connection or session exists between higher layer application entities only, and is invisible to UDP. As all UDP packets are independent of each other, a UDP packet header bears no information on whether it is a client to server or a server to client packet (in fact, UDP users are all equal in rights; the terms client and server cannot be defined explicitly). Therefore, the only recipe we can propose is to define as precisely as possible the range or set of those UDP port numbers which are allowed to communicate with the outer world.

Set filter which rejects whole UDP traffic passes through the "*rf5.0*" interface, but allows an interaction between internal and external DNS servers (port 53 is used to exchange data with the DNS server).

```
ipfw add accept udp from 0/0 53 to 0/0 53
ipfw add rf5.0 reject udp from 0/0 to 0/0
```