Manual:IP/Firewall/Mangle

From MikroTik Wiki
< Manual:IP | Firewall

Applies to RouterOS: v3, v4, v5, v6+

Contents

Summary

Sub-menu: `/ip firewall mangle`

Mangle is a kind of 'marker' that marks packets for future processing with special marks. Many other facilities in RouterOS make use of these marks, e.g. queue trees, NAT, routing. They identify a packet based on its mark and process it accordingly. The mangle marks exist only within the router, they are not transmitted across the network.

Additionally, the mangle facility is used to modify some fields in the IP header, like TOS (DSCP) and TTL fields.

Properties

Property	Description
`action` (*action name*; Default: **accept**)	Action to take if packet is matched by the rule: - `accept` - accept the packet. Packet is not passed to next firewall r - `add-dst-to-address-list` - add destination address to Address li specified by `address-list` parameter - `add-src-to-address-list` - add source address to Address list specified by `address-list` parameter - `change-dscp` - change Differentiated Services Code Point (DSCP) value specified by the new-dscp parameter - `change-mss` - change Maximum Segment Size field value of the pa to a value specified by the new-mss parameter - `change-ttl` - change Time to Live field value of the packet to a va specified by the new-ttl parameter - `clear-df` - clear 'Do Not Fragment' Flag - `jump` - jump to the user defined chain specified by the value of `jum target` parameter - `log` - add a message to the system log containing following data: interface, out-interface, src-mac, protocol, src-ip:port->dst-ip:port length of the packet. After packet is matched it is passed to next the list, similar as `passthrough`

	▪ `mark-connection` - place a mark specified by the new-connection- parameter on the entire connection that matches the rule ▪ `mark-packet` - place a mark specified by the new-packet-mark par on a packet that matches the rule ▪ `mark-routing` - place a mark specified by the new-routing-mark parameter on a packet. This kind of marks is used for policy routi purposes only ▪ `passthrough` - if packet is matched by the rule, increase counter a to next rule (useful for statistics). ▪ `return` - pass control back to the chain from where the jump took ▪ `route` - forces packets to a specific gateway IP by ignoring norma routing decision (prerouting chain only) ▪ `set-priority` - set priority specified by the new-priority parameter the packets sent out through a link that is capable of transporting priority (VLAN or WMM-enabled wireless interface). Read more> ▪ `sniff-pc` ▪ `sniff-tzsp` - send packet to a remote TZSP compatible system (s Wireshark). Set remote target with `sniff-target` and `sniff-targe port` parameters (Wireshark recommends port 37008) ▪ `strip-ipv4-options` - strip IPv4 option fields from IP header.
`address-list` (*string*; Default:)	Name of the address lis to be used. Applicable if action is `add-dst-to-address-list` or `add-src to-address-list`
`address-list-timeout` (*time*; Default: **00:00:00**)	Time interval after which the address will be removed from the addre list specified by `address list` parameter. Used in conjunction with `add-ds`

	`to-address-list` or `add-src-to-address-list` actions Value of `00:00:00` will leave the address in the address list forever
`chain` (*name*; Default:)	Specifies to which chain the rule will be added. If the input does not matc the name of an already defined chain, a new cha will be created.
`comment` (*string*; Default:)	Descriptive comment fo the rule.
`connection-bytes` (*integer-integer*; Default:)	Matches packets only if given amount of bytes h been transfered through the particular connection

	0 - means infinity, for example `connection-bytes=2000000-0` means that the rule matches if more than 2MB has bee transfered through the relevant connection
`connection-limit` (*integer,netmask*; Default:)	Restrict connection limi per address or address block
`connection-mark` (*no-mark | string*; Default:)	Matches packets marke via mangle facility with particular connection mark. If **no-mark** is set, rule will match any unmarked connection.
`connection-nat-state` (*srcnat | dstnat*; Default:)	Can match connections that are srcnatted,

	dstnatted or both. Note that connection-state=related connectio connection-nat-state is determined by direction the first packet. and if connection tracking nee to use dst-nat to deliver this connection to same hosts as main connectio it will be in connection-nat-state=dstnat even if there are no dst-nat rule at all.
`connection-rate` (*Integer 0..4294967295*; Default:)	Connection Rate is a firewall matcher that allows the capture of traffic based on the

	present speed of the connection. `Read more >`
`connection-state` (*estabilished | invalid | new | related*; Default:)	Interprets the connection tracking analysis data for a particular packe - `established` - a packet which belongs to an existing connection - `invalid` - a packet that does not have determined state in connec tracking (ussualy - sevear out-of-order packets, packets with wror sequence/ack number, or in case of resource overusage on route this reason invalid packet will not participate in NAT (as only connection-state=new packets do), and will still contain original s IP address when routed. We strongly suggest to drop all connecti state=invalid packets in firewall filter forward and input chains - `new` - the packet has started a new connection, or otherwise assoc with a connection which has not seen packets in both directions - `related` - a packet which is related to, but not part of an existing connection, such as ICMP errors or a packet which begins FTP da connection
`connection-type` (*ftp | h323 | irc | pptp | quake3 | sip | tftp*; Default:)	Matches packets from related connections bas on information from the connection tracking helpers. A relevant connection helper must enabled under /ip firewa service-port
`content` (*string*; Default:)	Match packets that contain specified text

`dscp` (*integer: 0..63*; Default:)	Matches DSCP IP heade field.
`dst-address` (*IP/netmask | IP range*; Default:)	Matches packets where destination is equal to specified IP or falls into specified IP range.
`dst-address-list` (*name*; Default:)	Matches destination address of a packet against user-defined address list
`dst-address-type` (*unicast | local | broadcast | multicast*; Default:)	Matches destination address type: ▪ `unicast` - IP address used for point to point transmission ▪ `local` - if dst-address is assigned to one of router's interfaces ▪ `broadcast` - packet is sent to all devices in subnet ▪ `multicast` - packet is forwarded to defined group of devices
`dst-limit` (*integer[/time],integer,dst-address | dst-port | src-address[/time]*; Default:)	Matches packets until a given pps limit is exceeded. As opposed to tl matcher, every destination IP address / destination port has it's own l Parameters are written in following format: `count[/time],burst,mode[/expire]`. ▪ **count** - maximum average packet rate measured in packets per `t` interval ▪ **time** - specifies the time interval in which the packet rate is measu (optional) ▪ **burst** - number of packets which are not counted by packet rate ▪ **mode** - the classifier for packet rate limiting ▪ **expire** - specifies interval after which recored ip address /port will deleted (optional)
`dst-port` (*integer[-integer]: 0..65535*; Default:)	

	List of destination port numbers or port numbe ranges
fragment (*yes|no*; Default:)	Matches fragmented packets. First (starting) fragment does not coun If connection tracking is enabled there will be no fragments as system automatically assemble every packet
hotspot (*auth | from-client | http | local-dst | to-client*; Default:)	
icmp-options (*integer:integer*; Default:)	Matches ICMP "type:cod fields
in-bridge-port (*name*; Default:)	Actual interface the packet has entered the router, if incoming interface is bridge
in-interface (*name*; Default:)	

	Interface the packet has entered the router
`ingress-priority` (*integer: 0..63*; Default:)	Matches ingress priority of the packet. Priority m be derived from VLAN, WMM or MPLS EXP bit. Read more >>
`ipsec-policy` (*in | out, ipsec | none*; Default:)	Matches the policy used by IpSec. Value is written in following forma **`direction, policy`**. Direction is Used to select whether to match the used for decapsulation or the policy that will be used for encapsulatio ▪ `in` - valid in the PREROUTING, INPUT and FORWARD chains ▪ `out` - valid in the POSTROUTING, OUTPUT and FORWARD chains ▪ `ipsec` - matches if the packet is subject to IpSec processing; ▪ `none` - matches packet that is not subject to IpSec processing (for example, IpSec transport packet). For example, if router receives Ipsec encapsulated Gre packet, then ru `ipsec-policy=in,ipsec` will match Gre packet, but rule `ipsec-policy=in,none` will match ESP packet.
`ipv4-options` (*any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp*; Default:)	Matches IPv4 header options. ▪ `any` - match packet with at least one of the ipv4 options ▪ `loose-source-routing` - match packets with loose source routing option. This option is used to route the internet datagram based o information supplied by the source ▪ `no-record-route` - match packets with no record route option. Th option is used to route the internet datagram based on informatio supplied by the source ▪ `no-router-alert` - match packets with no router alter option ▪ `no-source-routing` - match packets with no source routing option ▪ `no-timestamp` - match packets with no timestamp option ▪ `record-route` - match packets with record route option ▪ `router-alert` - match packets with router alter option ▪ `strict-source-routing` - match packets with strict source routin option ▪ `timestamp` - match packets with timestamp

`jump-target` (*name*; Default:)	Name of the target chai to jump to. Applicable o if `action=jump`
`layer7-protocol` (*name*; Default:)	Layer7 filter name defin in layer7 protocol menu.
`limit` (*integer,time,integer*; Default:)	Matches packets until a given pps limit is exceeded. Parameters are in following format: `count[/time],burst`. ▪ **count** - maximum average packet rate measured in packets per `t` interval ▪ **time** - specifies the time interval in which the packet rate is measu (optional, 1s will be used if not specified) ▪ **burst** - number of packets which are not counted by packet rate
`log-prefix` (*string*; Default:)	Adds specified text at th beginning of every log message. Applicable if `action=log`
`new-connection-mark` (*string*; Default:)	
`new-dscp` (*integer: 0..63*; Default:)	
`new-mss` (*integer*; Default:)	
`new-packet-mark` (*string*; Default:)	
`new-priority` (*integer*; Default:)	
`new-routing-mark` (*string*; Default:)	
`new-ttl` (*decrement | increment | set:integer*; Default:)	
`nth` (*integer,integer*; Default:)	Matches every nth pack `Read more >>`

`out-bridge-port` (*name*; Default:)	Actual interface the packet is leaving the router, if outgoing interface is bridge
`out-interface` (; Default:)	Interface the packet is leaving the router
`p2p` (*all-p2p | bit-torrent | blubster | direct-connect | edonkey | fasttrack | gnutella | soulseek | warez | winmx*; Default:)	Matches packets from various peer-to-peer (P2 protocols. Does not wor on encrypted p2p packe
`packet-mark` (*no-mark | string*; Default:)	Matches packets marke via mangle facility with particular packet mark. I **no-mark** is set, rule will match any unmarked packet.
`packet-size` (*integer[-integer]:0..65535*; Default:)	Matches packets of specified size or size

	range in bytes.
`passthrough` (*yes|no*; Default:)	whether to let the packe to pass further (like acti passthrough) after marking it with a given mark (property only vali action is mark packet, connection or routing mark).
`per-connection-classifier` (*ValuesToHash:Denominator/Remainder*; Default:)	PCC matcher allows division of traffic into equal streams with abili to keep packets with specific set of options i one particular stream. `Read more >>`
`port` (*integer[-integer]: 0..65535*; Default:)	Matches if any (source destination) port matche

	the specified list of port or port ranges. Applicab only if `protocol` is TCP o UDP
`protocol` (*name or protocol ID*; Default: **tcp**)	Matches particular IP protocol specified by protocol name or numbe
`psd` (*integer,time,integer,integer*; Default:)	Attempts to detect TCP and UDP scans. Parameters are in following `WeightThreshold, DelayThreshold, LopPortWeight, HighPortWeigh` - **WeightThreshold** - total weight of the latest TCP/UDP packets wi different destination ports coming from the same host to be treat port scan sequence - **DelayThreshold** - delay for the packets with different destination coming from the same host to be treated as possible port scan subsequence - **LowPortWeight** - weight of the packets with privileged (<=1024) destination port - **HighPortWeight** - weight of the packet with non-priviliged destina port
`random` (*integer: 1..99*; Default:)	Matches packets randomly with given probability.
`routing-mark` (*string*; Default:)	Matches packets marke by mangle facility with particular routing mark
`src-address` (*IP/Netmask, IP range*;	

Default:)	Matches packets where source is equal to specified IP or falls into specified IP range.
`src-address-list` (*name*; Default:)	Matches source addres of a packet against user defined address list
`src-address-type` (*unicast | local | broadcast | multicast*; Default:)	Matches source address type: ▪ `unicast` - IP address used for point to point transmission ▪ `local` - if address is assigned to one of router's interfaces ▪ `broadcast` - packet is sent to all devices in subnet ▪ `multicast` - packet is forwarded to defined group of devices
`src-port` (*integer[-integer]: 0..65535*; Default:)	List of source ports and ranges of source ports. Applicable only if protoc is TCP or UDP.
`src-mac-address` (*MAC address*; Default:)	Matches source MAC address of the packet
`tcp-flags` (*ack | cwr | ece | fin | psh | rst | syn | urg*; Default:)	Matches specified TCP flags ▪ `ack` - acknowledging data ▪ `cwr` - congestion window reduced ▪ `ece` - ECN-echo flag (explicit congestion notification) ▪ `fin` - close connection

	▪ `psh` - push function ▪ `rst` - drop connection ▪ `syn` - new connection ▪ `urg` - urgent data
`tcp-mss` (*integer: 0..65535*; Default:)	Matches TCP MSS value of an IP packet
`time` (*time-time,sat | fri | thu | wed | tue | mon | sun*; Default:)	Allows creation of a filte based on the packets' arrival time and date or, for locally generated packets, departure time and date
`tls-host` (*string*; Default:)	Allows to match traffic based on TLS hostname Accepts GLOB syntax (h ps://en.wikipedia.org/w i/Glob_(programming)) wildcard matching. Note that matcher will not be able to match hostname

	TLS handshake frame is fragmented into multiple TCP segments (packets
`ttl` (*equal | greater-than | less-than | not-equal : integer(0..255)*; Default:)	Matches packets TTL value.

Stats

`/ip firewall filter print stats` will show additional read-only properties

Property	Description
`bytes` (*integer*)	Total amount of bytes matched by the rule
`packets` (*integer*)	Total amount of packets matched by the rule

By default **print** is equivalent to **print static** and shows only static rules.

```
[admin@dzeltenais_burkaans] /ip firewall mangle> print stats
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN              ACTION                 BYTES           PACKETS
 0   prerouting         mark-routing           17478158        127631
 1   prerouting         mark-routing           782505          4506
```

To print also dynamic rules use **print all**.

```
[admin@dzeltenais_burkaans] /ip firewall mangle> print all stats
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN              ACTION                 BYTES           PACKETS
 0   prerouting         mark-routing           17478158        127631
 1   prerouting         mark-routing           782505          4506
 2 D forward            change-mss             0               0
 3 D forward            change-mss             0               0
 4 D forward            change-mss             0               0
 5 D forward            change-mss             129372          2031
```

Or to print only dynamic rules use **print dynamic**

```
[admin@dzeltenais_burkaans] /ip firewall mangle> print stats dynamic
Flags: X - disabled, I - invalid, D - dynamic
 #   CHAIN              ACTION                  BYTES           PACKETS
 0 D forward            change-mss              0               0
 1 D forward            change-mss              0               0
 2 D forward            change-mss              0               0
 3 D forward            change-mss              132444          2079
```

Menu specific commands

Property	Description
`reset-counters` (*id*)	Reset statistics counters for specified firewall rules.
`reset-counters-all` ()	Reset statistics counters for all firewall rules.

Basic examples

Change MSS

It is a well known fact that VPN links have smaller packet size due to encapsulation overhead. A large packet with MSS that exceeds the MSS of the VPN link should be fragmented prior to sending it via that kind of connection. However, if the packet has DF flag set, it cannot be fragmented and should be discarded. On links that have broken path MTU discovery (PMTUD) it may lead to a number of problems, including problems with FTP and HTTP data transfer and e-mail services.

In case of link with broken PMTUD, a decrease of the MSS of the packets coming through the VPN link solves the problem. The following example demonstrates how to decrease the MSS value via mangle:

```
/ip firewall mangle
add out-interface=pppoe-out protocol=tcp tcp-flags=syn action=change-mss new-mss=1300 chain=forward tcp-mss=1301-65535
```

Marking packets

Marking each packet is quite resource expensive especially if rule has to match against many parameters from IP header or address list containing hundreds of entries.

Lets say we want to

- mark all **tcp** packets except tcp/80 and match these packets against **first** address list
- mark all **udp** packets and match them against **second** address list.

```
/ip firewall mangle
  add chain=forward protocol=tcp port=!80 dst-address-list=first action=mark-packet new-packet-mark=first
  add chain=forward protocol=udp dst-address-list=second action=mark-packet new-packet-mark=second
```

Setup looks quite simple and probably will work without problems in small networks. Now multiply count of rules by 10, add few hundred entries in address list, run 100Mbit of traffic over this router and you will see how rapidly CPU usage is increasing. The reason for such behavior is that each rule reads IP header of every packet and tries to match collected data against parameters specified in firewall rule.

Fortunately if connection tracking is enabled, we can use connection marks to optimize our setup.

```
/ip firewall mangle
  add chain=forward protocol=tcp port=!80 dst-address-list=first connection-state=new action=mark-connection \
new-connection-mark=first
  add chain=forward connection-mark=first action=mark-packet new-packet-mark=first passthrough=no

  add chain=forward protocol=udp dst-address-list=second connection-state=new action=mark-connection \
new-connection-mark=second
  add chain=forward connection-mark=second action=mark-packet new-packet-mark=second passthrough=no
```

Now first rule will try to match data from IP header only from first packet of new connection and add connection mark. Next rule will no longer check IP header for each packet, it will just compare connection marks resulting in lower CPU consumption. Additionally `passthrough=no` was added that helps to reduce CPU consumption even more.

[Top | Back to Content **]**

Retrieved from "https://wiki.mikrotik.com/index.php?title=Manual:IP/Firewall/Mangle&oldid=30235"

Categories: Manual | Firewall

- This page was last edited on 2 January 2018, at 18:38.