Peer to Peer Traffic Control

Document revision 1.5 (Thu Sep 16 11:03:16 GMT 2004)
This document applies to V2.8

Table of Contents

General Information

Summary

This manual section describes techniques needed to control traffic from peer-to-peer (P2P) networks. Peer-to-peer is a concept whereby one individual host directly communicates with another, as opposed to each client referring to a common hub or server. This type of network connection allows users to share various information, including audio and video files and application programs. Uncontrolled P2P connections take all the available bandwith and left no space for other activities (like mail or HTTP browsing).

Quick Setup Guide

To drop all p2p application packets:

```
[admin@MikroTik] ip firewall rule forward> add action=drop p2p=all-p2p
```

Specifications

Packages required: ***system***
License required: ***level1 (Limited to 1 firewall rule), level3***
Home menu level: ***/ip firewall, /ip firewall mangle, /queue***
Hardware usage: *Increases with rule count*

Related Documents

- Firewall Filters
- Bandwidth Control
- Packet Marking (Mangle)

Description

RouterOS is able to recognize connections of the most popular P2P protocols:

- **Fasttrack** (Kazaa, KazaaLite, Diet Kazaa, Grokster, iMesh, giFT, Poisoned, mlMac)
- **Gnutella** (Shareaza, XoLoX, , Gnucleus, BearShare, LimeWire (java), Morpheus, Phex, Swapper, Gtk-Gnutella (linux), Mutella (linux), Qtella (linux), MLDonkey, Acquisition (Mac OS), Poisoned, Swapper, Shareaza, XoloX, mlMac)
- **Gnutella2** (Shareaza, MLDonkey, Gnucleus, Morpheus, Adagio, mlMac)
- **DirectConnect** (DirectConnect (AKA DC++), MLDonkey, NeoModus Direct Connect, BCDC++, CZDC++)
- **eDonkey** (eDonkey2000, eMule, xMule (linux), Shareaza, MLDonkey, mlMac, Overnet)
- **Soulseek** (Soulseek, MLDonkey)
- **BitTorrent** (BitTorrent, BitTorrent++, Shareaza, MLDonkey, ABC, Azureus, BitAnarch, SimpleBT, BitTorrent.Net, mlMac)
- **Blubster** (Blubster, Piolet)
- **WPNP** (WinMX)
- **Warez** (Warez; starting from 2.8.18)

Notes

The **Connection Tracking** facility (**/ip firewall connection tracking**) must be enabled if you want to track Peet-to-Peer protocols.

It is impossible to recognize peer-to-peer traffic from the first packet. Only already established connections can be matched. That also means that in case source NAT is treating Peer-to-Peer traffic differently from the regular traffic, Peer-to-Peer programs will not work (general application is policy-routing redirecting regular traffic through one interface and Peer-to-Peer traffic - through another)

The filter will work only if it sees the traffic coming from both directions.

Traffic Marking

Home menu level: ***/ip firewall mangle***

Description

Peer-to-peer traffic marking provided by **Mangle** facility labels the traffic for future processing aginst the firewall filters or queues.

Property Description

mark-connection (*text*; default: "") - change connection mark of the packet to this value

mark-flow (*text*; default: "") - change flow mark of the packet to this value

p2p (*any* | *all-p2p* | *bit-torrent* | *direct-connect* | *fasttrack* | *soulseek* | *blubster* | *edonkey* | *gnutella* | *warez*; default: **any**) - match Peer-to-Peer (P2P) connections:

- **all-p2p** - match all known P2P traffic
- **any** - match any packet (i.e., do not check this property)

Traffic Filtering

Home menu level: ***/ip firewall***

Description

RouterOS gives you ability to filter out traffic generated by P2P networks.

Property Description

action (*accept* | *drop* | *jump* | *passthrough* | *reject* | *return*; default: **accept**) - ation to undertake if the packet matches the rule, one of the:

- **accept** - accept the packet. No action, i.e., the packet is passed through without undertaking any action, except for mangle, and no more rules are processed in the relevant list/chain
- **drop** - silently drop the packet (without sending the ICMP reject message)
- **jump** - jump to the chain specified by the value of the jump-target argument
- **passthrough** - ignore this rule, except for mangle, go on to the next one. Acts the same way as a disabled rule, except for ability to count and mangle packets
- **reject** - reject the packet and send an ICMP reject message
- **return** - return to the previous chain, from where the jump took place

connection (*text*; default: "") - connection mark to match. Only connections (including related) marked in the MANGLE would be matched

flow (*text*) - flow mark to match. Only packets marked in the MANGLE would be matched

jump-target (*name*) - name of the target chain, if the action=jump is used

p2p (*any* | *all-p2p* | *bit-torrent* | *direct-connect* | *fasttrack* | *soulseek* | *blubster* | *edonkey* | *gnutella* | *warez*; default: **any**) - match Peer-to-Peer (P2P) connections:

- **all-p2p** - match all known P2P traffic
- **any** - match any packet (i.e., do not check this property)

Traffic Limiting

Home menu level: ***/queue***

Description

You can limit peer-to-peer traffic to a given number of Kbits per second or give it lower priority than, for example HTTP traffic.

It is also possible to prioritize small file downloading over large ones using queue bursts.

Application Examples

Summary

This section will give you two examples of typical peer-to-peer traffic control configurations.

Cumulative Bandwith Limiting

Consider the following example:

Suppose we need to drop all the P2P traffic coming from the Internet, but allow the use of WinMX client between two offices limiting it to 284 Kbps in both directions. You need to do the following:

- Allow WinMX client to be used between two offices

```
[admin@MikroTik] ip firewall rule forward> add p2p=winmx action=accept
src-address=10.0.0.0/24 dst-address=10.0.1.0/24
[admin@MikroTik] ip firewall rule forward> add p2p=winmx action=accept
dst-address=10.0.0.0/24 src-address=10.0.1.0/24
```

- Drop all other P2P traffic

```
[admin@MikroTik] ip firewall rule forward> add p2p=all-p2p action=drop
```

- Limit the traffic to 284 Kbps

```
[admin@MikroTik] queue simple> add dst-address=10.0.1.0/24 max-limit=290816/290816
```

Per Address Queuing

Suppose we want to limit each P2P user to a given amount of Kbps. This can be done on a per-address basis.

We should define custom queue type **kind=pcq** to accomplish the task. Each user upload and download rates would be limited to the **pcq-rate** value in the relevant queue.

- First we need to mark the P2P traffic:

```
[admin@MikroTik] ip firewall mangle> add src-address=10.0.0.0/24 mark-flow=p2p-out \
\... p2p=all-p2p action=passthrough
[admin@MikroTik] ip firewall mangle> add dst-address=10.0.0.0/24 mark-flow=p2p-in \
\... p2p=all-p2p action=passthrough
[admin@MikroTik] ip firewall mangle>
```

- Then create custom queue type with **kind=pcq**:

```
[admin@MikroTik] queue type> add name="p2p-out" kind=pcq \
\... pcq-rate=65536 pcq-classifier=src-address
```

```
[admin@MikroTik] queue type> add name="p2p-in" kind=pcq pcq-rate=65536 \
\... pcq-classifier=dst-address
[admin@MikroTik] queue type>
```

- Finally, add two queues to the queue tree:

```
[admin@MikroTik] queue tree> add name="p2p-in" \
\... parent=global-in flow=p2p-in queue=p2p-in
[admin@MikroTik] queue tree> add name="p2p-out" \
\... parent=global-out flow=p2p-out queue=p2p-out
[admin@MikroTik] queue tree>
```