

SOCKS Proxy Server

Document revision 1.4 (January 14, 2008, 11:23 GMT)

This document applies to V3.0

Table of Contents

[Table of Contents](#)

[Summary](#)

[Specifications](#)

[Description](#)

[Notes](#)

[Additional Documents](#)

[SOCKS Configuration](#)

[Description](#)

[Property Description](#)

[Example](#)

[Access List](#)

[Description](#)

[Property Description](#)

[Active Connections](#)

[Description](#)

[Property Description](#)

[Example](#)

[FTP service through SOCKS server](#)

General Information

Summary

This manual discusses the SOCKS proxy server which is implemented in RouterOS. MikroTik RouterOS supports SOCKS version 4.

Specifications

Packages required: *system*

License required: *level1*

Home menu level: */ip socks*

Standards and Technologies: [SOCKS version 4](#)

Hardware usage: *Not significant*

Description

SOCKS is a proxy server that allows TCP based application data to relay across the firewall, even if the firewall would block the packets. The SOCKS protocol is independent from application protocols, so it can be used for many services, e.g, WWW, FTP, TELNET, and others.

At first, an application client connects to the SOCKS proxy server, then the proxy server looks in its **access**

list to see whether the client is permitted to access the remote application resource or not, if it is permitted, the proxy server relies the packet to the application server and creates a connection between the application server and client.

Notes

Remember to configure your application client to use SOCKS version 4.

You should secure the SOCKS proxy using its access list and/or firewall to disallow access from outside. Failing to secure the proxy server may introduce security issues to your network, and may provide a way for spammers to send junk mail through the router.

Additional Documents

- [Information about SOCKS](#)

SOCKS Configuration

Description

In this section you will learn how to enable the SOCKS proxy server and do its configuration.

Property Description

connection-idle-timeout (*time*; default: **2m**) - time after which idle connections are terminated

enabled (yes | no; default: **no**) - whether to enable or no the SOCKS proxy

max-connections (*integer*: 1..500; default: **200**) - maximum number of simultaneous connections

port (*integer*: 1..65535; default: **1080**) - TCP port on which the SOCKS server listens for connections

Example

To enable SOCKS:

```
[admin@MikroTik] ip socks> set enabled=yes
[admin@MikroTik] ip socks> print
      enabled: yes
      port: 1080
connection-idle-timeout: 2m
      max-connections: 200
[admin@MikroTik] ip socks>
```

Access List

Home menu level: */ip socks access*

Description

In the SOCKS access list you can add rules which will control access to SOCKS server. This list is similar to

firewall lists.

Property Description

action (*allow* | *deny*; default: **allow**) - action to be performed for this rule

- **allow** - allow packets, matching this rule, to be forwarded for further processing
- **deny** - deny access for packets, matching this rule

dst-address (*IP addressnetmask*) - destination (server's) address

dst-port (*port*) - destination TCP port

src-address (*IP addressnetmask*) - source (client's) address for a packet

src-port (*port*) - source TCP port

Active Connections

Home menu level: */ip socks connections*

Description

The Active Connection list shows all established TCP connections, which are maintained through the SOCKS proxy server.

Property Description

dst-address (*read-only: IP address*) - destination (application server) IP address

rx (*read-only: integer*) - bytes received

src-address (*read-only: IP address*) - source (application client) IP address

tx (*read-only: integer*) - bytes sent

type (*read-only: in* | *out* | *unknown*) - connection type

- **in** - incoming connection
- **out** - outgoing connection
- **unknown** - connection has just been initiated

Example

To see current TCP connections:

```
[admin@MikroTik] ip socks connections> print
# SRC-ADDRESS DST-ADDRESS TX RX
0 192.168.0.2:3242 159.148.147.196:80 4847 2880
1 192.168.0.2:3243 159.148.147.196:80 3408 2127
2 192.168.0.2:3246 159.148.95.16:80 10172 25207
3 192.168.0.2:3248 194.8.18.26:80 474 1629
4 192.168.0.2:3249 159.148.95.16:80 6477 18695
5 192.168.0.2:3250 159.148.95.16:80 4137 27568
6 192.168.0.2:3251 159.148.95.16:80 1712 14296
7 192.168.0.2:3258 80.91.34.241:80 314 208
8 192.168.0.2:3259 80.91.34.241:80 934 524
9 192.168.0.2:3260 80.91.34.241:80 930 524
10 192.168.0.2:3261 80.91.34.241:80 312 158
11 192.168.0.2:3262 80.91.34.241:80 312 158
```

```
[admin@MikroTik] ip socks connections>
```

Application Examples

FTP service through SOCKS server

Let us consider that we have a network **192.168.0.0/24** which is masqueraded, using a router with a public IP **10.1.0.104/24** and a private IP **192.168.0.1/24**. Somewhere in the network is an FTP server with IP address **10.5.8.8**. We want to allow access to this FTP server for a client in our local network with IP address **192.168.0.2/24**.

We have already masqueraded our local network:

```
[admin@MikroTik] ip firewall nat> print
Flags: X - disabled, I - invalid, D - dynamic
0 chain=srcnat action=masquerade src-address=192.168.0.0/24
[admin@MikroTik] ip firewall nat>
```

And the access to public FTP servers is denied in firewall:

```
[admin@MikroTik] ip firewall filter> print
Flags: X - disabled, I - invalid, D - dynamic
0 chain=forward action=drop src-address=192.168.0.0/24 dst-port=21 protocol=tcp
[admin@MikroTik] ip firewall filter>
```

We need to enable the SOCKS server:

```
[admin@MikroTik] ip socks> set enabled=yes
[admin@MikroTik] ip socks> print
enabled: yes
port: 1080
connection-idle-timeout: 2m
max-connections: 200
[admin@MikroTik] ip socks>
```

Add access to a client with an IP address **192.168.0.2/32** to SOCKS access list, allow data transfer from FTP server to client (allow destination ports from 1024 to 65535 for any IP address), and drop everything else:

```
[admin@MikroTik] ip socks access> add src-address=192.168.0.2 dst-port=21 \
\... action=allow
[admin@MikroTik] ip socks access> add dst-port=1024-65535 action=allow
[admin@MikroTik] ip socks access> add action=deny
[admin@MikroTik] ip socks access> print
Flags: X - disabled
0 src-address=192.168.0.2 dst-port=21 action=allow
1 dst-port=1024-65535 action=allow
2 action=deny
[admin@MikroTik] ip socks access>
```

That's all - the SOCKS server is configured. To see active connections and data transmitted and received:

```
[admin@MikroTik] ip socks connections> print
# SRC-ADDRESS DST-ADDRESS TX RX
0 192.168.0.2:1238 10.5.8.8:21 1163 4625
1 192.168.0.2:1258 10.5.8.8:3423 0 3231744
[admin@MikroTik] ip socks connections>
```

Note! In order to use SOCKS proxy server, you have to specify its IP address and port in your FTP client. In this case IP address would be **192.168.0.1** (local IP address of the router/SOCKS server) and TCP port **1080**.