HTTP Proxy

Document revision 1.2 (Tue May 23 14:34:47 GMT 2006)
This document applies to V2.9

Table of Contents

General Information

Summary

The MikroTik RouterOS implements the following proxy server features:

- Regular HTTP proxy
- Transparent proxy. Can be transparent and regular at the same time
- Access list by source, destination, URL and requested method
- Cache access list (specifies which objects to cache, and which not)
- Direct Access List (specifies, which resources should be accessed directly, and which - through another proxy server)
- Logging facility

Quick Setup Guide

To enable HTTP proxy, do the following:

```
[admin@MikroTik] ip proxy> set enabled=yes
[admin@MikroTik] ip proxy> print
                   enabled: yes
                 src-address: 0.0.0.0
                        port: 8080
                parent-proxy: 0.0.0.0:0
                 cache-drive: system
          cache-administrator: "webmaster"
          max-disk-cache-size: none
           max-ram-cache-size: 100000KiB
           cache-only-on-disk: yes
  maximal-client-connections: 1000
  maximal-server-connections: 1000
              max-object-size: 2000KiB
               max-fresh-time: 3d
[admin@MikroTik] ip proxy>
```

Remember to secure your proxy by preventing unauthorized access to it, otherwise it may be used as an open proxy. Also you need to setup destination NAT in order to utilize transparent proxying facility:

```
[admin@MikroTik] ip firewall nat> add chain=dstnat protocol=tcp dst-port=80
action=redirect to-ports=8080
[admin@MikroTik] ip firewall nat> print
Flags: X - disabled, I - invalid, D - dynamic
 0   chain=dstnat protocol=tcp dst-port=80 action=redirect to-ports=8080
[admin@MikroTik] ip firewall nat>
```

Specifications

Packages required: ***system***
License required: ***level3***
Home menu level: ***/ip proxy***
Standards and Technologies: ***HTTP/1.0***, ***HTTP/1.1***, ***FTP***

Related Documents

- Software Package Management

- IP Addresses and ARP
- Log Management

Description

This service performs proxying of HTTP and HTTP-proxy (for FTP, HTTP and HTTPS protocols) requests. Web proxy performs Internet object cache function by storing requested Internet objects, i.e., data available via HTTP and FTP protocols on a system positioned closer to the recipient than the site the data is originated from. Here 'closer' means increased path reliability, speed or both. Web browsers can then use the local proxy cache to speed up access and reduce bandwidth consumption.

When setting up proxy service, make sure it serves only your clients, and is not misused as relay. Please read the security notice in the Access List Section!

Note that it may be useful to have Web proxy running even with no cache when you want to use it only as something like HTTP and FTP firewall (for example, denying access to mp3 files) or to redirect requests to external proxy (possibly, to a proxy with caching functions) transparently.

Setup

Home menu level: ***/ip proxy***

Property Description

cache-administrator (*text*; default: **webmaster**) - administrator's e-mail displayed on proxy error page

cache-drive (*system* | *name*; default: **system**) - specifies the target disk drive to be used for storing cached objects. You can use console completion to see the list of available drives

cache-only-on-disk (yes | no; default: **yes**) - whether to create database in memory that describes cache contents on disk. This will minimize memory consumption, but may affect speed

enabled (yes | no; default: **no**) - whether the proxy server is enabled

max-disk-cache-size (*none* | *unlimited* | *integer*: 0..4294967295; default: **none**) - specifies the maximal disk cache size, measured in kibibytes

max-fresh-time (*time*; default: **3d**) - maximal time to store a cached object. The validity period of an object is is usually defined by the object itself, but in case it is set too high, you can override the maximal value

maximal-client-connecions (*integer*; default: **1000**) - maximal number of connections accepted from clients (any further connections will be rejected)

maximal-server-connectons (*integer*; default: **1000**) - maximal number of connections made to servers (any further connections from clients will be put on hold until some server connections will terminate)

max-object-size (*integer*; default: **2000KiB**) - objects larger than the size specified will not be saved on disk. The value is measured in kibibytes. If you wish to get a high bytes hit ratio, you should probably increase this (one 2 MiB object hit counts for 2048 1KiB hits). If you wish to increase speed more than your want to save bandwidth you should leave this low

max-ram-cache-size (*none* | *unlimited* | *integer*: 0..4294967295; default: **none**) - specifies the

maximal RAM cache size, measured in kibibytes

parent-proxy (*IP address* | *port*; default: **0.0.0.0:0**) - IP address and port of another HTTP proxy to redirect all requests to (exceptions may be defined in the "direct access" list

- **0.0.0.0:0** - no parent proxy is used

port (*port*; default: **8080**) - TCP port the proxy server will be listening on. This is to be specified on all clients that want to use the server as HTTP proxy. Transparent (with zero configuration for clients) proxy setup can be made by redirecting HTTP requests to this port in IP firewall using destination NAT feature

src-address (*IP address*; default: **0.0.0.0**) - the web-proxy will use this address connecting to the parent proxy or web site.

- **0.0.0.0** - appropriate src-address will be automatically taken from the routing table

Notes

The web proxy listens to all IP addresses that the router has in its IP address list.

Example

To enable the proxy on port 8000:

```
[admin@MikroTik] ip proxy> set enabled=yes port=8000
[admin@MikroTik] ip proxy> print
                    enabled: yes
                src-address: 0.0.0.0
                       port: 8000
               parent-proxy: 0.0.0.0:0
                cache-drive: system
        cache-administrator: "dmitry@mikrotik.com"
        max-disk-cache-size: none
         max-ram-cache-size: 100000KiB
         cache-only-on-disk: yes
  maximal-client-connections: 1000
  maximal-server-connections: 1000
            max-object-size: 2000KiB
             max-fresh-time: 3d
[admin@MikroTik] ip proxy>
```

Access List

Home menu level: ***/ip proxy access***

Description

Access list is configured like a regular firewall rules. Rules are processed from the top to the bottom. First matching rule specifies decision of what to do with this connection. There is a total of 6 classifiers that specify matching constraints. If none of these classifiers is specified, the particular rule will match every connection.

If connection is matched by a rule, **action** property of this rule specifies whether connection will be allowed or not. If the particular connection does not match any rule, it will be allowed.

Property Description

action (*allow* | *deny*; default: **allow**) - specifies whether to pass or deny matched packets

dst-address (*IP address* | *netmask*) - destination address of the IP packet

dst-host (*wildcard*) - IP address or DNS name used to make connection the target server (this is the string user wrote in his/her browser before specifying port and path to a particular web page)

dst-port (*port*) - a list or range of ports the packet is destined to

hits (*read-only: integer*) - the number of requests that were policed by this rule

local-port (*port*) - specifies the port of the web proxy via which the packet was received. This value should match one of the ports web proxy is listening on.

method (*any* | *connect* | *delete* | *get* | *head* | *options* | *post* | *put* | *trace*) - HTTP method used in the request (see HTTP Methods section in the end of this document)

path (*wildcard*) - name of the requested page within the target server (i.e. the name of a particular web page or document without the name of the server it resides on)

redirect-to (*text*) - in case access is denied by this rule, the user shall be redirected to the URL specified here

src-address (*IP address* | *netmask*) - source address of the IP packet

Notes

Wildcard properties (**dst-host** and **dst-path**) match a complete string (i.e., they will not match "example.com" if they are set to "example"). Available wildcards are '*' (match any number of any characters) and '?' (match any one character). Regular expressions are also accepted here, but if the property should be treated as a regular expression, it should start with a colon (':').

Small hits in using regular expressions:

- \\ symbol sequence is used to enter \ character in console
- **\.** pattern means **.** only (in regular expressions single dot in pattern means any symbol)
- to show that no symbols are allowed before the given pattern, we use ^ symbol at the beginning of the pattern
- to specify that no symbols are allowed after the given pattern, we use **$** symbol at the end of the pattern
- to enter **[** or **]** symbols, you should escape them with backslash \.

It is strongly recommended to deny all IP addresses except those behind the router as the proxy still may be used to access your internal-use-only (intranet) web servers. Also, consult examples in Firewall Manual on how to protect your router.

Direct Access List

Home menu level: ***/ip proxy direct***

Description

If **parent-proxy** property is specified, it is possible to tell proxy server whether to try to pass the request to the parent proxy or to resolve it connecting to the requested server directly. Direct Access List is managed

just like Proxy Access List described in the previous chapter except the **action** argument.

Property Description

action (*allow* | *deny*; default: **allow**) - specifies the action to perform on matched packets

- **allow** - always resolve matched requests directly bypassing the parent router
- **deny** - resolve matched requests through the parent proxy. If no one is specified this has the same effect as allow

dst-address (*IP address* | *netmask*) - destination address of the IP packet

dst-host (*wildcard*) - IP address or DNS name used to make connection the target server (this is the string user wrote in his/her browser before specifying port and path to a particular web page)

dst-port (*port*) - a list or range of ports the packet is destined to

hits (*read-only: integer*) - the number of requests that were policed by this rule

local-port (*port*) - specifies the port of the web proxy via which the packet was received. This value should match one of the ports web proxy is listening on.

method (*any* | *connect* | *delete* | *get* | *head* | *options* | *post* | *put* | *trace*) - HTTP method used in the request (see HTTP Methods section in the end of this document)

path (*wildcard*) - name of the requested page within the target server (i.e. the name of a particular web page or document without the name of the server it resides on)

src-address (*IP address* | *netmask*) - source address of the IP packet

Notes

Unlike the access list, the direct proxy access list has default action equal to **deny**. It takes place when no rules are specified or a particular request did not match any rule.

Cache Management

Home menu level: ***/ip web-proxy cache***

Description

Cache access list specifies, which requests (domains, servers, pages) have to be cached locally by web proxy, and which not. This list is implemented exactly the same way as web proxy access list. Default action is to cache object (if no matching rule is found).

Property Description

action (*allow* | *deny*; default: **allow**) - specifies the action to perform on matched packets

- **allow** - cache objects from matched request
- **deny** - do not cache objects from matched request

dst-address (*IP address* | *netmask*) - destination address of the IP packet

dst-host (*wildcard*) - IP address or DNS name used to make connection the target server (this is the string user wrote in his/her browser before specifying port and path to a particular web page)

dst-port (*port*) - a list or range of ports the packet is destined to

hits (*read-only: integer*) - the number of requests that were policed by this rule

local-port (*port*) - specifies the port of the web proxy via which the packet was received. This value should match one of the ports web proxy is listening on.

method (*any* | *connect* | *delete* | *get* | *head* | *options* | *post* | *put* | *trace*) - HTTP method used in the request (see HTTP Methods section in the end of this document)

path (*wildcard*) - name of the requested page within the target server (i.e. the name of a particular web page or document without the name of the server it resides on)

src-address (*IP address* | *netmask*) - source address of the IP packet

Proxy Monitoring

Command name: ***/ip proxy monitor***

Description

This command displays some stats of the proxy server

Property Description

cache-used (*read-only: integer*) - disk space used for the cache

hits (*read-only: integer*) - number of requests found in cache and served from there

hits-sent-to-clients (*read-only: integer*) - amount of data served from the cache

ram-cache-used (*read-only: integer*) - RAM space used to store the cache

received-from-servers (*read-only: integer*) - amount of data received from other servers

requests (*read-only: integer*) - number of requests handled

sent-to-clients (*read-only: integer*) - amount of data sent to the clients of this proxy server

status (*read-only: text*; default: **stopped**) - display status information of the proxy server

- **stopped** - proxy is disabled and is not running
- **rebuilding-cache** - proxy is enabled and running, existing cache is being verified
- **running** - proxy is enabled and running
- **stopping** - proxy is shutting down (max 10s)
- **clearing-cache** - proxy is stopped, cache files are being removed
- **creating-cache** - proxy is stopped, cache directory structure is being created
- **dns-missing** - proxy is enabled, but not running because of unknown DNS server (you should specify it under /ip dns)
- **invalid-address** - proxy is enabled, but not running because of invalid address (you should change address or port)
- **invalid-cache-administrator** - proxy is enabled, but not running because of invalid cache-administrator's e-mail address
- **invalid-hostname** - proxy is enabled, but not running because of invalid hostname (you should set a valid hostname value)
- **error-logged** - proxy is not running because of unknown error. This error is logged as

System-Error. Please, send us this error and some description, how it happened

- **reserved-for-cache (integer)** - maximal cache size, that is accessible to web-proxy

total-ram-used (*read-only: integer*) - total amount of RAM used for the proxy

uptime (*read-only: time*) - the time since the proxy has been started last time

Connection List

Home menu level: ***/ip proxy connections***

Description

This menu conntains the list of current connections the proxy is serving

Property Description

dst-address (*read-only: IP address*) - IP address of the connection

protocol (*read-only: text*) - protocol name

rx-bytes (*read-only: integer*) - the amount of bytes received by the client

src-address (*read-only: IP address*) - IP address of the connection originator

state (*read-only: closing* | *connecting* | *converting* | *hotspot* | *idle* | *resolving* | *rx-header* | *tx-body* | *tx-eof* | *tx-header* | *waiting*) - opened connection state

- **closing** - the data transfer is finished, and the connection is being finalized
- **connecting** - establishing toe connection
- **converting** - replacing header and footer fields in response or request paket
- **hotspot** - check if hotspot authentication allows to continue (for hotspot proxy)
- **idle** - staying idle
- **resolving** - resolving server's DNS name
- **rx-header** - receiving HTTP header
- **tx-body** - transmitting HTTP body to the client
- **tx-eof** - writing chunk-end (when converting to chunked response)
- **tx-header** - transmitting HTTP header to the client
- **waiting** - waiting for transmission form a peer

tx-bytes (*read-only: integer*) - the amount of bytes sent by the client

Cache inserts

Home menu level: ***/ip proxy inserts***

Description

This menu shows statistics on objects stored in cache (cache inserts)

Property Description

denied (*read-only: integer*) - number of inserts denied by the caching list

errors (*read-only: integer*) - number of disk or other system-related errors

no-memory (*read-only: integer*) - number of objects not stored because there was not enough memory

successes (*read-only: integer*) - number of successfull cache inserts

too-large (*read-only: integer*) - number of objects too large to store

Cache Lookups

Home menu level: ***/ip proxy lookups***

Description

This menu shows statistics on objects read from cache (cache lookups)

Property Description

denied (*read-only: integer*) - number of requests denied by the access list

expired (*read-only: integer*) - number of requests found in cache, but expired, and, thus, requested from an external server

no-expiration-info (*read-only: integer*) - conditional request received for a page that does not have the information to compare the request with

non-cacheable (*read-only: integer*) - number of requests requested from the external servers unconditionally (as their caching is denied by the cache access list)

not-found (*read-only: integer*) - number of requests not found in the cache, and, thus, requested from an external server (or parent proxy if configured accordingly)

successes (*read-only: integer*) - number of requests found in the cache

Complementary Tools

Home menu level: ***/ip proxy***

Description

Web proxy has additional commands to handle non-system drive used for caching purposes and to recover the proxy from severe file system errors.

Command Description

check-drive - checks non-system cache drive for errors

clear-cache - deletes existing cache and creates new cache directories

format-drive - formats non-system cache drive and prepairs it for holding the cache

HTTP Methods

Description

OPTIONS

This method is a request of information about the communication options available on the chain between the client and the server identified by the **Request-URI**. The method allows the client to determine the options and (or) the requirements associated with a resource without initiating any resource retrieval

GET

This method retrieves whatever information identified by the **Request-URI**. If the **Request-URI** refers to a data processing process than the response to the **GET** method should contain data produced by the process, not the source code of the process procedure(-s), unless the source is the result of the process.

The **GET** method can become a *conditional* **GET** if the request message includes an **If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match**, or **If-Range** header field. The conditional **GET** method is used to reduce the network traffic specifying that the transfer of the entity should occur only under circumstances described by conditional header field(-s).

The **GET** method can become a *partial* **GET** if the request message includes a **Range** header field. The partial **GET** method intends to reduce unnecessary network usage by requesting only parts of entities without transferring data already held by client.

The response to a **GET** request is cacheable if and only if it meets the requirements for HTTP caching.

HEAD

This method shares all features of **GET** method except that the server must not return a message-body in the response. This retrieves the metainformation of the entity implied by the request which leads to a wide usage of it for testing hypertext links for validity, accessibility, and recent modification.

The response to a **HEAD** request may be cacheable in the way that the information contained in the response may be used to update previously cached entity identified by that **Request-URI**.

POST

This method requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the **Request-URI**.

The actual action performed by the **POST** method is determined by the origin server and usually is **Request-URI** dependent.

Responses to **POST** method are not cacheable, unless the response includes appropriate **Cache-Control** or **Expires** header fields.

PUT

This method requests that the enclosed entity be stored under the supplied **Request-URI**. If another entity exists under specified **Request-URI**, the enclosed entity should be considered as updated (newer) version

of that residing on the origin server. If the **Request-URI** is not pointing to an existing resource, the origin server should create a resource with that URI.

If the request passes through a cache and the **Request-URI** identifies one or more currently cached entities, those entries should be treated as stale. Responses to this method are not cacheable.

TRACE

This method invokes a remote, application-layer loop-back of the request message. The final recipient of the request should reflect the message received back to the client as the entity-body of a 200 (OK) response. The final recipient is either the origin server or the first proxy or gateway to receive a **Max-Forwards** value of **0** in the request. A **TRACE** request must not include an entity.

Responses to this method MUST NOT be cached.