Web Proxy

Document revision 1.2 (Tue May 16 14:04:40 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 set up a 1 GiB large web cache which will listen on port 8000, do the following:

```
[admin@MikroTik] ip web-proxy> set enabled=yes port=8000 max-cache-size=1048576
[admin@MikroTik] ip web-proxy> print
                   enabled: yes
               src-address: 0.0.0.0
                      port: 8000
                  hostname: proxy
         transparent-proxy: no
              parent-proxy: 0.0.0.0:0
       cache-administrator: webmaster
           max-object-size: 4096 KiB
               cache-drive: system
            max-cache-size: 1048576 KiB
        max-ram-cache-size: unlimited
                    status: rebuilding-cache
        reserved-for-cache: 9216 KiB
    reserved-for-ram-cache: 2048 KiB
[admin@MikroTik] ip web-proxy>
```

Remember to secure your proxy by preventing unauthorized access to it, otherwise it may be used as an open proxy.

Specifications

Packages required: ***web-proxy***
License required: ***level3***
Home menu level: ***/ip web-proxy***
Standards and Technologies: ***HTTP/1.0***, ***HTTP/1.1***, ***FTP***
Hardware usage: *uses memory and disk space, if available (see description below)*

Related Documents

- Software Package Management
- IP Addresses and ARP
-
- Log Management

Description

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 Web proxy, 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 as something like HTTP and FTP firewall (for example, denying access to mp3 files) or to redirect requests to external proxy transparently.

Setup

Home menu level: ***/ip web-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

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

hostname (*text*; default: **proxy**) - hostname (DNS or IP address) of the web proxy

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

max-object-size (*integer*; default: **4096**) - 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: **unlimited**) - specifies the maximal memory cache size, measured in kibibytes

parent-proxy (*IP address* | *port*; default: **0.0.0.0:0**) - specifies upper-level (parent) proxy

port (*port*; default: **3128**) - specifies the port(s) the web proxy will be listening on

reserved-for-cache (*read-only: integer*; default: **0**) - specifies allocated memory cache size, measured in kibibytes

reserved-for-ram-cache (*read-only: integer*; default: **2048**) - specifies allocated memory cache size, measured in kibibytes

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

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

transparent-proxy (yes | no; default: **no**) - specifies whether the proxy uses transparent mode or not

Notes

By default the proxy cache can use as much disk space as there is allocated for it. When the system allocates the space for the proxy cache, 1/7th of the total partition (disk) size is reserved for the system, but not less than 50MB. The rest is left for the proxy cache. The system RAM size is considered as well when allocating the cache size. The cache size is limited so, that there are at least 15MB of RAM per 1GB of cache plus 55MB of RAM is reserved for the system. **max-cache-size** is also taken in account, so the cache will not occupy more than it is specified in this property. The effective limit is calculated as a minimum of all three limits. Note also that RouterOS supports up to 950MB of memory.

Considering the previous note, you should be aware that you will not be able to enable web proxy, if you have less than 60MB of RAM on your router

Expire time of cache entries can be different for each HTML page (specified in headers). But, if there is no such header, the entry will be considered fresh for not more than 72 hours.

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

Example

To enable the proxy on port 8080:

```
[admin@MikroTik] ip web-proxy> set enabled=yes port=8080
[admin@MikroTik] ip web-proxy> print
                   enabled: yes
               src-address: 0.0.0.0
                      port: 8080
                  hostname: proxy
         transparent-proxy: no
              parent-proxy: 0.0.0.0:0
       cache-administrator: webmaster
           max-object-size: 4096 KiB
               cache-drive: system
             max-cache-size: none
        max-ram-cache-size: unlimited
                    status: running
        reserved-for-cache: 0 KiB
    reserved-for-ram-cache: 2048 KiB
[admin@MikroTik] ip web-proxy>
```

Access List

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

Description

Access list is configured in the same way as MikroTik RouterOS 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.

By default, there is one rule, which prevents **connect** requests to ports other then **443** and **563**.

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

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 at the end of this document)

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

url (*wildcard*) - the URL of the HTTP request

Notes

There is one rule by default, that disallows **connect** method connections to ports other than **443** (https) and **563** (snews). **connect** method is a security hole that allows connections (transparent tunneling) to any computer using any protocol. It is used mostly by spammers, as they found it very convenient to use others' mail (SMTP) servers as anonymous mail relay to send spam over the Internet.

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.

Wildcard property **url** matches 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 \.

Example

The default rule:

```
[admin@MikroTik] ip web-proxy access> print
Flags: X - disabled, I - invalid
 0   ;;; allow CONNECT only to SSL ports 443 [https] and 563 [snews]
     dst-port=!443,563 method=connect action=deny
[admin@MikroTik] ip web-proxy access>
```

To disallow download of .MP3 and .MPG files and FTP connections other than from the **10.0.0.1** server:

```
[admin@MikroTik] ip web-proxy access> add url=":\\.mp\[3g\]$" action=deny
[admin@MikroTik] ip web-proxy access> add src-address=10.0.0.1/32 action=allow
[admin@MikroTik] ip web-proxy access> add url="ftp://*" action=deny
[admin@MikroTik] ip web-proxy access> print
Flags: X - disabled, I - invalid
 0   ;;; allow CONNECT only to SSL ports 443 [https] and 563 [snews]
     dst-port=!443,563 method=connect action=deny

 1   url=":\.mp[3g]$" action=deny

 2   src-address=10.0.0.1/32 action=allow

 3   url="ftp://*" action=deny
[admin@MikroTik] ip web-proxy access>
```

Direct Access List

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

Description

If **parent-proxy** property is specified, it is possible to tell the 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-port (*port*) - a list or range of ports the packet is destined to

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)

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

url (*wildcard*) - the URL of the HTTP request

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

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)

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

url (*wildcard*) - the URL of the HTTP request

Complementary Tools

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

Transparent Mode

Description

Transparent proxy feature performs request caching invisibly to the end-user. This way the user does not notice that his connection is being processed by the proxy and therefore does not need to perform any additional configuration of the software he is using.

This feature may as well be combined with bridge to simplify deployment of web proxy in the existing infrastructure.

To enable the transparent mode, place a firewall rule in destination NAT, specifying which connections, *id est* traffic coming to which ports should be redirected to the proxy.

Notes

Only HTTP traffic is supported in transparent mode of the web proxy. HTTPS and FTP protocols are not going to work this way.

Example

To configure the router to transparently redirect all connections coming from **ether1** interface to port **80** to the web proxy listening on port **8080**, then add the following destination NAT rule:

```
[admin@MikroTik] > /ip firewall nat add in-interface=ether1 dst-port=80 \
\... protocol=tcp action=redirect to-ports=8080 chain=dstnat
[admin@MikroTik] > /ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0   chain=dstnat protocol=tcp in-interface=ether1 dst-port=80 action=redirect
     to-ports=8080
[admin@MikroTik] >
```

Be aware, that you will not be able to access the router's web page after addition of the rule above unless you will change the port for the **www** service under **/ip service** submenu to a different value or explicitly exclude router's IP address from those to be matched, like:

It is assumed that the router's address is **1.1.1.1/32**.

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.