HotSpot Gateway

Document revision 3.4 (Sat May 29 07:57:03 GMT 2004)
This document applies to MikroTik RouterOS V2.8

Table of Contents

Table of Contents
General Information
  Summary
  Specifications
  Related Documents
  Description
Question&Answer-Based Setup
  Command Description
  Notes
  Example
HotSpot Gateway Setup
  Property Description
  Command Description
  Notes
  Example
HotSpot User Profiles
  Description
  Property Description
  Notes
  Example
HotSpot Users
  Property Description
  Notes
  Example
HotSpot Active Users
  Description
  Property Description
  Example
HotSpot Remote AAA
  Property Description
  Notes
  Example
HotSpot Server Settings
  Description
  Property Description
  Notes
  Example
HotSpot Cookies
  Description
  Property Description
  Notes

Example
Walled Garden
Description
Property Description
Notes
Example
Customizing HotSpot Servlet
Description
Notes
Example
Possible Error Messages
Description
HotSpot Step-by-Step User Guide for dhcp-pool Method
Description
Example
HotSpot Step-by-Step User Guide for enabled-address Method
Description
Example
Optional Settings

General Information

Summary

The MikroTik HotSpot Gateway enables providing of public network access for clients using wireless or wired network connections.

HotSpot Gateway features:

- authentication of clients using local client database, or RADIUS server
- accounting using local database, or RADIUS server
- Walled-garden system (accessing some web pages without authorization)
- HotSpot Gateway can provide access for authorized clients using two different methods:
  - **dhcp-pool** method uses DHCP server to assign temporary (not valid in outer networks) IP addresses to clients prior to authentication. After successful authentication the DHCP server assigns an IP address to the client from a different IP pool. This method may be used to assign a fixed IP address to each user (i.e. no matter which computer does the user use, he/she will always use the same IP address)
  - **enabled-address** method enables traffic for authorized clients without need of IP address change
- traffic and connection time accounting
- clients can be limited by:
  - download/upload speed (tx/rx bitrate)
  - connection time
  - downloaded/uploaded traffic (bytes)

Universal Client feature may be used with HotSpot enabled-address method to provide IP network services regardless of client computers' IP network settings

Specifications

Packages required: ***hotspot, dhcp (optional)***
License required: ***level1 (Limited to 1 active user), level3 (Limited to 1 active user), level4 (Limited to 200 active users), level5 (Limited to 500 active users), level6***
Home menu level: ***/ip hotspot***
Standards and Technologies: ***ICMP***, ***DHCP***
Hardware usage: ***Not significant***

Related Documents

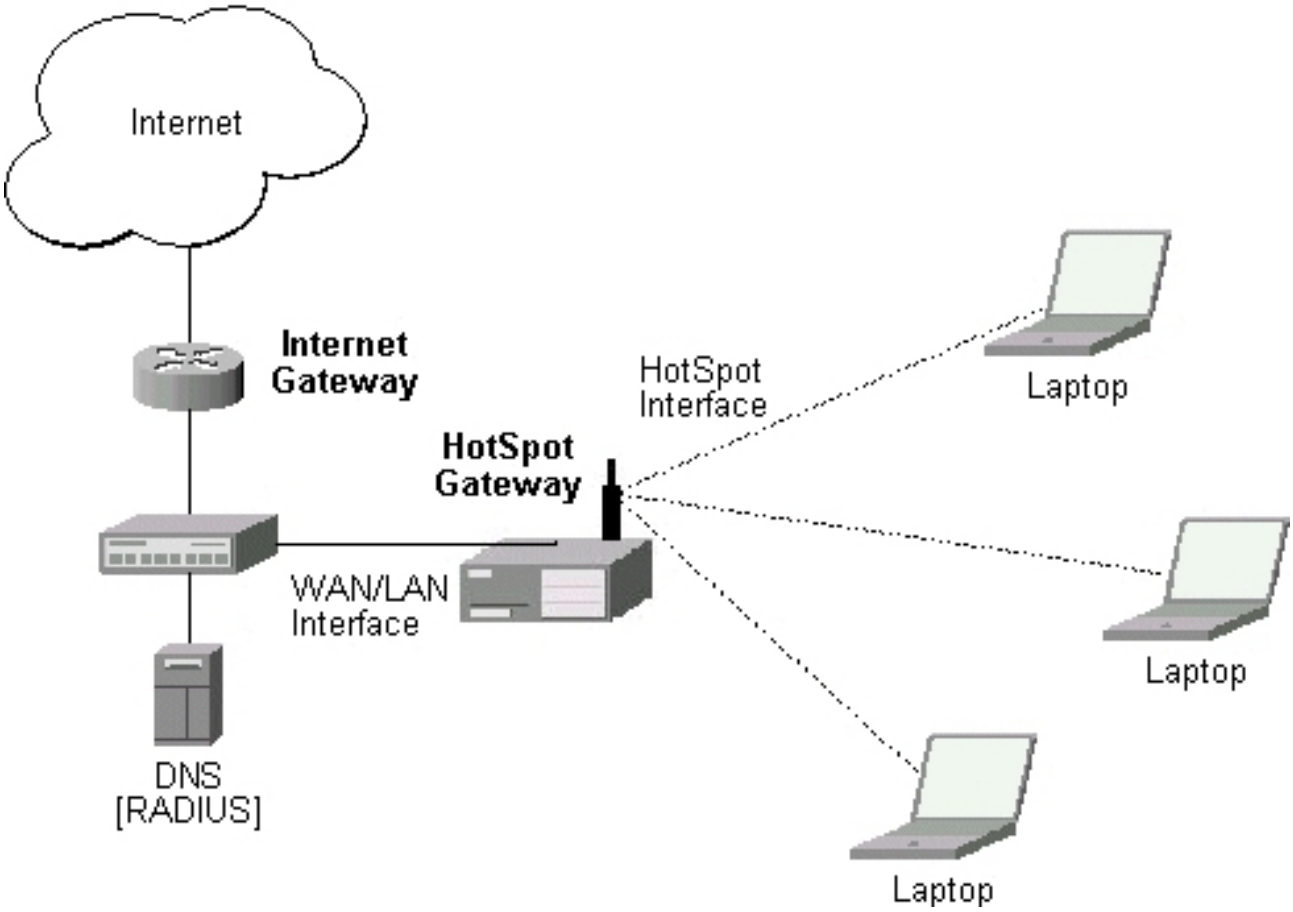
- ***Package Management***
- ***IP Addresses and ARP***
- ***IP Pools***
- ***DHCP Client and Server***
- ***AAA***
- ***Firewall Filters***
- ***Packet Marking (Mangle)***
- ***Network Address Translation***
- ***Connection Tracking and Service Ports***

Description

MikroTik HotSpot Gateway should have at least two network interfaces:

1. HotSpot interface, which is used to connect HotSpot clients
2. LAN/WAN interface, which is used to access network resources. For example, DNS and RADIUS server(s) should be accessible

The diagram below shows a sample HotSpot setup.



The HotSpot interface should have an IP address assigned to it. To use **dhcp-pool** method, there should be two IP addresses: one as the gateway for the temporary IP address pool used prior to authentication, and second as the gateway for the permanent IP address pool used by authenticated clients. **Note**, that you have to provide routing for these address pools, unless you plan to use masquerading (source NAT). Physical network connection has to be established between the HotSpot user's computer and the gateway. It can be wireless (the wireless card should be registered to AP), or wired (the NIC card should be connected to a hub or a switch).

In **dhcp-pool** case, the arp mode of the HotSpot interface should be set to **reply-only** to prevent network access using static IP addresses (the DHCP server should add static ARP entries for each DHCP client). **Note** also that Universal Client feature can not be used with **dhcp-pool** method.

Introduction to HotSpot

HotSpot is a way to authorize users to access some network resources. It does not provide traffic encryption. To log in, users may use almost any web browser (either HTTP or HTTPS protocol), so they are not required to install additional software. The gateway is accounting the uptime and amount of traffic each of its clients have used, and also can send this information to a RADIUS server. The HotSpot system may limit each particular user's bitrate, total amount of traffic, uptime and some other parameters mentioned further in this document.

The HotSpot system is targeted to provide authentication within a local network, but may as well be used to authorize access from outer networks to local networks. Configuring firewall rules, it is possible to exclude some IP networks and protocols from authentication and/or accounting. The

walled garden feature allows users to access some web pages without the need of prior authentication.

HotSpot system is rather simple by itself, but it must be used in conjunction with other features of RouterOS. Using many RouterOS features together it is possible to make a Plug-and-Play access system.

There are two login methods for HotSpot users - **dhcp-pool** and **enabled-address**. The **enabled-address** is the preferred one in most cases, but if you want to bind together usernames and IP addresses (i.e. if you want a user to get the same IP address no matter which computer is he/she using), then the **dhcp-pool** method is the only possibility.

The Initial Contact

First, a client gets an IP address. It may be set statically or be given out by a DHCP server. If the client tries to access network resources using a web browser, the destination NAT rule redirects that TCP connection request to the HotSpot servlet (TCP port 8088 for HTTP by default; HTTPS may also be used on its default TCP port 443). This brings up the HotSpot Welcome/Login page where the user should input his/her username and password (the page may be customized as described later on).

It is very important to understand that login method for a particular user is determined only after the user is authenticated and no assumptions are made by the router before.

Walled Garden

It is possilbe to specify a number of domains which can be accessed without prior registration. This feature is called Walled Garden. When a not logged-in user sends a HTTP request to an allowed web page, the HotSpot gateway redirects the request to the original destination (or to a specified parent proxy). When a user is logged in, there is no effect of this table for him/her.

To implement the Walled Garden feature an embedded web proxy server has been designed, so all the requests from not authorized users are really going through this proxy. **Note** that the embedded proxy server does not have caching function yet. Also note that this embedded proxy server is in the **hotspot** software package and does not require **web-proxy** package.

Authentication

In case of HTTP protocol, HotSpot servlet generates an MD5 hash challenge to be used together with the user's password for computing the string which will be sent to the HotSpot gateway. The hash result together with username is sent over network to HotSpot service (so, password is never sent in plain text over IP network). On the client side, MD5 algorithm is implemented in JavaScript applet, so if a browser does not support JavaScript (like, for example, Internet Explorer 2.0 or some PDA browsers), it will not be able to authenticate users. It is possible to allow unencrypted passwords to be accepted, but it is not recommended to use this feature.

If HTTPS protocol is used, HotSpot user just send his/her password without additional hashing. In either case, HTTP POST method (if not possible, then - HTTP GET method) is used to send data to the HotSpot gateway.

HotSpot can authenticate users using local user database or a RADIUS server (local database is consulted first, then - a RADIUS server). If authentication is done locally, profile corresponding to

that user is used, otherwise (in case of RADIUS) default profile is used to set default values for parameters, which are not set in RADIUS access-accept message. For more information on how the interaction with a RADIUS server works, see the respective manual section.

If authentication by HTTP cookie is enabled, then after each successful login cookie is sent to web browser and the same cookie is added to active HTTP cookie list. Next time a user will try to log in, web browser will send http cookie. This cookie will be compared to the one stored on the HotSpot gateway and only if there is the same source MAC address and the same randomly generated ID, user will be automatically logged in. Otherwise, the user will be prompted to log in, and in the case authentication was successfull, old cookie will be removed from the local HotSpot active cookie list and the new one with different random ID and expiration time will be added to the list and sent to the web browser.

RADIUS authentication is CHAP by defalt, but it is possible to force the HotSpot gateway to use PAP. To do this, you should enable unencrypted passwords, and remove the possibility for the servlet to hash the passwords (see **Customizing HotSpot servlet** chapter on how to do it).

Authorization

One of the two login methods is to be used for each client individually (you may choose one or allow it to be done automatically in user profile configuration). The **enabled-address** method is the preferred one, so if it is configured correctly and the client has a proper IP address (that matches the one set in the user database), this method will be used. If the **enabled-address** method is not enabled or the client's IP address should be changed, the HotSpot Gateway tries to use **dhcp-pool** method. In that case, MikroTik HotSpot Gateway's DHCP server tries to change the DHCP address lease the client might have received before the authentication. It is possible to specify what IP addresses each particular user will receive after he/she logs in (that way a user will always get the same IP no matter what computer he/she has logged in from)

Address assignment with dhcp-pool login method

To create a HotSpot infrastructure with **dhcp-pool** method, DHCP server should be configured to lease IP addresses from a temporary IP address pool for a very short period of time (lease time at about 14 seconds; lesser values may cause problems with some DHCP clients). This temporary subnet should have some restrictions, so that the users received a temporary IP address could only access the HotSpot login page.

Once a user is authenticated, the HotSpot gateway changes the lease assigned to the user so that he/she will receive an IP address from a different IP address pool when the lease time of the current temporary lease will be over (it is not possible to recall DHCP lease, so the address will only change when the temporary lease expires).

Accounting

The HotSpot system makes user accounting through firewall rules. You should create a **hotspot** firewall chain, and the system will put there two dynamic rules for each active user (one for upload, and one for download). You shold make all the traffic you need accounting for to pass through this firewall table.

Question&Answer-Based Setup

Command name: ***/ip hotspot setup***

Questions

hotspot interface (*name*) - interface to run HotSpot on

interface already configured (yes | no; default: **no**) - whether to add hotspot authentication for the existing interface setup or otherwise interface setup should be configured from the scratch

enable universal client (yes | no; default: **no**) - whether to enable Universal Client on the HotSpot interface

login method (*dhcp-pool* | *enabled-address* | *smart*; default: **enabled-address**) - login method to use

local address of temporary network (*IP address*; default: **192.168.0.1/24**) - temporary HotSpot address for the interface (for dhcp-pool method)

masquerade temporary network (yes | no; default: **yes**) - whether to masquerade the temporary network

address pool of temporary network (*name*) - IP address pool the for temporary HotSpot network

local address of hotspot network (*IP address*; default: **10.5.50.1/24**) - HotSpot address for the interface

masquerade hotspot network (yes | no; default: **yes**) - whether to masquerade the HotSpot network

address pool of hotspot network (*name*) - IP address pool for the HotSpot network

use ssl (yes | no; default: **no**) - whether to use secure SSL authentication

import and setup certificate (yes | no; default: **yes**) - if the setup should try to import and set up a certificate

passphrase (*text*) - the passphrase of the certificate

select certificate (*name*) - which certificate to use

another port for service (*integer*; default: **4430**) - if there is already a service on the 443 TCP port, setup will move that service on an another port, so that HotSpot secure authentication page would be on standard port for SSL

ip address of smtp server (*IP address*; default: **0.0.0.0**) - IP address of the SMTP server to redirect SMTP requests (TCP port 25) to

- **0.0.0.0** - no redirect

use transparent web proxy (yes | no; default: **no**) - whether to use transparent web proxy for hotspot clients

use local dns cache (yes | no) - whether to redirect all DNS requests (UDP port 53) to the local DNS cache

dns servers (*IP address* | *IP address*) - DNS servers for HotSpot clients

dns name (*text*) - DNS domain name of the HotSpot gateway

another port for service (*integer*; default: **8081**) - another port for www service (so that hotspot service could be put on port 80)

name of local hotspot user (*text*; default: **admin**) - username of one automatically created user

password for the user (*text*) - password for the automatically created user

Notes

Depending on current settings and answers to the previous questions, default values of following questions may be different. Some questions may disappear if they become redundant (for example, there is no use of setting up temporary network when login method is **enabled-address**)

If Universal Client is enabled, and DNS cache is not used, DNS requests are redirected to the first DNS server configured.

Example

To configure HotSpot on ether1 interface (which is already configured), enabling transparent web proxy and adding user admin with password rubbish:

```
[admin@MikroTik] ip hotspot> setup
Select interface to run HotSpot on

hotspot interface: ether1
Use SSL authentication?

use ssl: no
Add hotspot authentication for existing interface setup?

interface already configured: yes
Create local hotspot user

name of local hotspot user: admin
password for the user: rubbish
Use transparent web proxy for hotspot clients?

use transparent web proxy: yes
[admin@MikroTik] ip hotspot>
```

HotSpot Gateway Setup

Home menu level: ***/ip hotspot***

Property Description

allow-unencrypted-passwords (yes | no; default: **no**) - whether to authenticate user if plain-text password is received

auth-http-cookie (yes | no; default: **no**) - defines whether HTTP authentication by cookie is enabled

auth-mac (yes | no; default: **no**) - defines whether authentication by Ethernet MAC address is enabled

auth-mac-password (yes | no; default: **no**) - use MAC address as a password if MAC authorization is enabled

auth-requires-mac (yes | no; default: **yes**) - whether to require client's IP address to resolve to MAC address (i.e. whether to require that all the clients are in the same Ethernet-like network (as opposed to IP network, Ethernet-like network is bounded by routers) as the HotSpot gateway)

dns-name (*text*) - DNS name of the HotSpot server

hotspot-address (*IP address*; default: **0.0.0.0**) - IP address for HotSpot service (used for www

access)

http-cookie-lifetime (*time*; default: **1d**) - validity time of HTTP cookies

login-mac-universal (yes | no; default: **no**) - whether to log in every host of Universal client instantly in case it has its MAC address listed in HotSpot user list

parent-proxy (*IP address*; default: **0.0.0.0**) - the address of the proxy server the HotSpot service will use as a parent proxy

split-user-domain (yes | no; default: **no**) - whether to split username from domain name when the username is given in "user@domain" or in "domain\user" format

status-autorefresh (*time*; default: **1m**) - WWW status page autorefresh time

universal-proxy (yes | no; default: **no**) - whether to intercept the requests to HTTP proxy servers

use-ssl (yes | no; default: **no**) - whether the servlet allows only HTTPS:

- **yes** - the registration may only occur using the Secure HTTP (HTTPS) protocol
- **no** - the registration may be accomplished using both HTTP and HTTPS protocols

Command Description

reset-html - overwrite the existing HotSpot servlet with the original HTML files. It is used if you have changed the servlet and it is not working after that.

Notes

If **dns-name** property is not specified, **hotspot-address** is used instead. If **hotspot-address** is also absent, then both are to be detected automatically.

If **auth-mac** is enabled, then a client is not prompted for username and password if the MAC address of this computer is in the HotSpot user database (either local or on RADIUS). Nevertheless this method does not excuse clients from the common login procedure, just from filling out the registration form (i.e. regardless of whether MAC authorization is applicable for a client, he/she should open the Login page in order to get registered). The only exception is the users of Universal Client - if **login-mac-universal** property is enabled, they will not even have to open a web browser if their MAC addresses are listed in the user database.

The **universal-proxy** feature automatically creates DST-NAT rules to redirect requests of each particular user to a proxy server he/she is using (it may be set in his/her settings to use an unknown to us proxy server) to the local embedded proxy server. This feature may be used in combination with Universal Client feature to provide Internet access for users regardless of their network settings.

allow-unencrypted-passwords property makes it possible to authenticate with the browsers not supporting JavaScript (for example, Internet Explorer 2.0 or some PDA browsers). It is also possible to log in using telnet connection, just requesting the page `/login?user=username&password=password`. An another use of this property is the possibility of hard-coded authentication information in the servlet's login page simply creating the appropriate link.

auth-requires-mac property makes it possible to make a 'reverse HotSpot' - to authenticate users accessing the local network from the Internet.

Example

To enable cookie support:

```
[admin@MikroTik] ip hotspot> set auth-http-cookie=yes
[admin@MikroTik] ip hotspot> print
                        use-ssl: no
                hotspot-address: 0.0.0.0
                        dns-name: ""
              status-autorefresh: 1m
                 universal-proxy: no
                    parent-proxy: 0.0.0.0:0
               auth-requires-mac: yes
                        auth-mac: no
               auth-mac-password: no
                auth-http-cookie: yes
            http-cookie-lifetime: 1d
    allow-unencrypted-passwords: no
             login-mac-universal: no
              split-user-domain: no
[admin@MikroTik] ip hotspot>
```

HotSpot User Profiles

Home menu level: ***/ip hotspot profile***

Description

HotSpot User profiles are used for common user settings. Profiles are like user groups, they are grouping users with the same limits.

Property Description

idle-timeout (*time*; default: **0s**) - idle timeout (maximal period of inactivity) for client

- **0** - no timeout

incoming-filter (*name*) - name of the firewall chain applied to incoming packets

keepalive-timeout (*time*; default: **2m**) - keepalive timeout for client

- **0** - no timeout

login-method - the login method user will be using

- **dhcp-pool** - login by changing IP address via DHCP server
- **enabled-address** - login by enabling access for client's existing IP address
- **smart** - choose best login method for each case

mark-flow (*name*) - traffic from authorized users will be marked by firewall mangle with this flow name

name (*name*) - profile reference name

outgoing-filter (*name*) - name of the firewall chain applied to outgoing packets

rx-bit-rate (*integer*; default: **0**) - receive bitrate (for users it is upload bitrate)

- **0** - no limitation

session-timeout (*time*; default: **0s**) - session timeout (maximal session time) for client

- **0** - no timeout

shared-users (*integer*; default: **1**) - maximal number of simultaneously logged in users with the same username

tx-bit-rate (*integer*; default: **0**) - transmit bitrate (for users it is download bitrate)

- **0** - no limitation

Notes

To use **enabled-address** method, **mark-flow** should be set. To use **dhcp-pool** method, **dhcp** software package must be installed

idle-timeout is used to detect, that client is not using outer networks (e.g. Internet), i.e., there is NO TRAFFIC coming from that client and going through the router. **keepalive-timeout** is used to detect, that the computer of the client is still alive and reachable. If check will fail during this period, client will be logged out. **session-timeout** is an unconditional uptime limit

To choose the login method to be used if **smart** method is set as the value of **login-method** property, the following algorithm is used:

- If a client has a dynamic DHCP address lease received from the router, correct HotSpot server is set for the DHCP server issued that lease, and the client has specific IP address set in the **/ip hotspot user** configuration, **dhcp-pool** method will be used
- else, if **mark-flow** property is defined in the client's profile), **enabled-address**method will be used
- else, if the client has a dynamic DHCP lease, **dhcp-pool**method will be used
- else, an error message will be displayed, and the client will not be logged in

Example

To use **enabled-address** method that uses **logged-in** mark and logs a client off if he disappears for more then a minute:

```
[admin@MikroTik] ip hotspot profile> set default login-method=enabled-address \
\... mark-flow=logged-in keepalive-timeout=1m
[admin@MikroTik] ip hotspot profile> print
Flags: * - default
  0 * name="default" session-timeout=0s idle-timeout=0s only-one=yes
      tx-bit-rate=0 rx-bit-rate=0 incoming-filter="" outgoing-filter=""
      mark-flow="logged-in" login-method=enabled-address keepalive-timeout=1m

[admin@MikroTik] ip hotspot profile>
```

To define an additional profile that will also limit download speed to 64 kilobyte/s and upload data rate to 32 kilobyte/s, and call it **limited**:

```
[admin@MikroTik] ip hotspot profile> add copy-from=default tx-bit-rate=65536 \
\... rx-bit-rate=32768 name=limited
[admin@MikroTik] ip hotspot profile> print
Flags: * - default
  0 * name="default" session-timeout=0s idle-timeout=0s only-one=yes
      tx-bit-rate=0 rx-bit-rate=0 incoming-filter="" outgoing-filter=""
      mark-flow="logged-in" login-method=enabled-address keepalive-timeout=1m
```

```
  1   name="limited" session-timeout=0s idle-timeout=0s only-one=yes
      tx-bit-rate=65536 rx-bit-rate=32768 incoming-filter=""
      outgoing-filter="" mark-flow="logged-in" login-method=enabled-address
      keepalive-timeout=1m

[admin@MikroTik] ip hotspot profile>
```

HotSpot Users

Home menu level: ***/ip hotspot user***

Property Description

address (*IP address*; default: **0.0.0.0**) - static IP address. If not 0.0.0.0, client will always get the same IP address. It implies, that only one simultaneous login for that user is allowed

bytes-in (*read-only: integer*) - total amount of bytes received from user

bytes-out (*read-only: integer*) - total amount of bytes sent to user

limit-bytes-in (*integer*; default: **0**) - maximum amount of bytes user can transmit

- **0** - no limit

limit-bytes-out (*integer*; default: **0**) - maximum amount of bytes user can receive

- **0** - no limit

limit-uptime (*time*; default: **0s**) - total uptime limit for user (pre-paid time)

- **0s** - no limit

mac-address (*MAC address*; default: **00:00:00:00:00:00**) - static MAC address. If not 00:00:00:00:00:00, client is allowed to login only from that MAC address

name (*name*) - user name

packets-in (*read-only: integer*) - total amount of packets received from user

packets-out (*read-only: integer*) - total amount of packets sent to user

password (*text*) - user password

profile (*name*; default: **default**) - user profile

routes (*text*) - routes that are to be registered on the HotSpot gateway when the client is connected. The route format is: "dst-address gateway metric" (for example, "10.1.0.0/24 10.0.0.1 1"). Several routes may be specified separated with commas

uptime (*read-only: time*) - total time user has been logged in

Notes

If **auth-mac** property is enabled, clients' MAC addresses (written with CAPITAL letters) can be used as usernames. If **auth-mac-password** is set to **no**, there should be no password for that users. Otherwise, the password should be equal to the username. When a client is connecting, his/her MAC address is checked first. If there is a user with that MAC address, the client is authenticated as this user. If there is no match, client is asked for username and password.

The **address** property is used only for **dhcp-pool** login method to tell it DHCP server. If a user already has a permanent IP address (as it is happening when **enabled-address** method is used), this

property will just be ignored.

The byte limits are total limits for each user (not for each session as at **/ip hotspot active**). So, if a user has already downloaded something, then session limit will show the total limit - (minus) already downloaded. For example, if download limit for a user is 100MB and the user has already downloaded 30MB, then session download limit after login at **/ip hotspot active** will be 100MB - 30MB = 70MB.

Should a user reach his/her limits (bytes-in >= limit-bytes-in or bytes-out >= limit-bytes-out), he/she will not be able to log in anymore.

The statistics is updated if a user is authenticated via local user database each time he/she logs out. It means, that if a user is currently logged in, then the statistics will not show current total values. Use **/ip hotspot active** submenu to view the statistics on the current user sessions.

Example

To add user Ex with password Ex that is allowed to log in only with 01:23:45:67:89:AB MAC address and is limited to 1 hour of work:

```
[admin@MikroTik] ip hotspot user> add name=Ex password=Ex \
\... mac-address=01:23:45:67:89:AB limit-uptime=1h
[admin@MikroTik] ip hotspot user> print
Flags: X - disabled
  #   NAME        ADDRESS         MAC-ADDRESS       PROFILE     UPTIME
  0   Ex          0.0.0.0         01:23:45:67:89:AB default     0s
[admin@MikroTik] ip hotspot user> print detail
Flags: X - disabled
  0   name="Ex" password="Ex" address=0.0.0.0 mac-address=01:23:45:67:89:AB
      profile=default routes="" limit-uptime=1h limit-bytes-in=0
      limit-bytes-out=0 uptime=0s bytes-in=0 bytes-out=0 packets-in=0
      packets-out=0

[admin@MikroTik] ip hotspot user>
```

HotSpot Active Users

Home menu level: ***/ip hotspot active***

Description

The active user list shows the list of currently logged in users. Nothing can be changed here, except user can be logged out with the **remove** command

Property Description

address (*read-only: IP address*) - IP address of the user

bytes-in (*read-only: integer*) - how many bytes did the router receive from the client

bytes-out (*read-only: integer*) - how many bytes did the router send to the client

domain (*read-only: text*) - domain of the user (if split from username)

idle-timeout (*read-only: time*) - how much idle time it is left for the user until he/she will be automatically logged out

keepalive-lost (*read-only: time*) - how much time past since last packed from the client has been received

packets-in (*read-only: integer*) - how many packets did the router receive from the client

packets-out (*read-only: integer*) - how many packets did the router send to the client

session-timeout (*read-only: time*) - how much time is left for the user until he/she will be automatically logged out

uptime (*read-only: time*) - current session time (logged in time) of the user

user (*read-only: name*) - name of the user

Example

To get the list of active users:

```
[admin@MikroTik] ip hotspot active> print
Flags: R - radius, H - DHCP
  #    USER        ADDRESS         UPTIME          SESSION-TIMEOUT IDLE-TIMEOUT
  0    Ex          10.0.0.144      4m17s           55m43s
[admin@MikroTik] ip hotspot active>
```

HotSpot Remote AAA

Home menu level: ***/ip hotspot aaa***

Property Description

accounting (yes | no; default: **yes**) - whether RADIUS accounting should be used (have no effect if RADIUS is not used)

interim-update (*time*; default: **0s**) - Interim-Update time interval

- **0s** - do not send accounting updates

use-radius (yes | no; default: **no**) - whether user database in a RADIUS server should be consulted

Notes

RADIUS user database is consulted only if the required username is not found in local user database

The value set in **interim-update** is overridden by the value sent by a RADIUS server (if any)

Example

To enable RADIUS AAA:

```
[admin@MikroTik] ip hotspot aaa> set use-radius=yes
[admin@MikroTik] ip hotspot aaa> print
        use-radius: yes
        accounting: yes
    interim-update: 0s
[admin@MikroTik] ip hotspot aaa>
```

HotSpot Server Settings

Home menu level: ***/ip hotspot server***

Description

HotSpot Server configuration is used to modify DHCP leases for logged-in users in order them to get non-temporary addresses. When a user has successfully authenticated, the HotSpot Server communicates with the DHCP server to change the lease information the user will receive next time he/she will request the DHCP lease (that is why the lease of temporary address should be as short as possible). The new lease should not be for a long time either for users to be able to switch fast on one machine as well as to reuse the IP addresses of this pool (users are logged out just as they click the log out button, but their addresses stay allocated to the machines they have been using, making it impossible for another users to log in from these machines)

Property Description

address-pool (*name*) - IP pool name, from which a HotSpot client will get an IP address if it is not given a static IP address

dhcp-server (*name*) - DHCP server with which to use this profile

lease-time (*time*; default: **1m**) - DHCP lease time for logged in user

login-delay (*time*; default: **10s**) - Time required to log user in. The after-login page is displayed for this time. This time should be approximately the same as the lease-time for the temporary address lease

name (*name*) - server profile name

Notes

This configuration is ignored by **enabled-address** method.

There can be added one HotSpot Server for each DHCP server. Which server profile to apply will depend on DHCP server which gave DHCP lease to that client. Actually it means that if user will log in from different interfaces, then different server profiles will be used. It allows assigning different IP addresses on different Ethernet interfaces.

Network mask, gateway and some other setting are set up in **/ip dhcp network** submenu

Example

To add a HotSpot server named **dhcp1** to the DHCP server **hotspot-dhcp** giving IP addresses from the **hotspot** address pool:

```
[admin@MikroTik] ip hotspot server> add name=dhcp1 dhcp-server=hotspot-dhcp \
\... address-pool=hotspot
[admin@MikroTik] ip hotspot server> print
  # NAME                 DHCP-SERVER    ADDRESS-POOL LOGIN-DELAY  LEASE-TIME
  0 dhcp1                hotspot-dhcp   hotspot      10s          1m

[admin@MikroTik] ip hotspot server>
```

HotSpot Cookies

Home menu level: ***/ip hotspot cookie***

Description

Cookies can be used for authentication in the Hotspot service

Property Description

domain (*read-only: text*) - domain name (if split from username)

expires-in (*read-only: time*) - how long the cookie is valid

mac-address (*read-only: MAC address*) - user's MAC address

user (*read-only: name*) - username

Notes

There can be multiple cookies with the same MAC address. For example, there will be a separate cookie for each web browser on the same computer.

Cookies can expire - that's the way how it is supposed to be. Default validity time for cookies is **1** day (24 hours), but it can be changed:

```
/ip hotspot set http-cookie-lifetime=3d
```

Example

To get the list of valid cookies:

```
[admin@MikroTik] ip hotspot cookie> print
  # USER               DOMAIN            MAC-ADDRESS       EXPIRES-IN
  0 Ex                                   01:23:45:67:89:AB 23h54m16s
[admin@MikroTik] ip hotspot cookie>
```

Walled Garden

Home menu level: ***/ip hotspot walled-garden***

Description

Walled garden is a system which allows unauthorized use of some resources, but requires authorization to access other resources. This is useful, for example, to give access to some general information about HotSpot service provider or billing options.

Property Description

action (*allow* | *deny*; default: **allow**) - action to undertake if a packet matches the rule:

- **allow** - allow the access to the page without prior authorization

- **deny** - the authorization is required to access this page

dst-host (*text*; default: "") - domain name of the destination web server (this is regular expression)

dst-port (*integer*; default: "") - the TCP port a client has send the request to

path (*text*; default: "") - the path of the request (this is regular expression)

Notes

Currently you can not place HTTPS servers inside the Walled Garden. However, there is a workaround on this. You can add a mangle rule that allows you to pass traffic to an IP address of secure web server, *exempli gratia*:

```
/ip firewall mangle add dst-address=159.148.108.1/32 mark-flow=hs-auth
```

Example

To allow unauthorized requests to the **www.example.com** domain's **/paynow.html** page:

```
[admin@MikroTik] ip hotspot walled-garden> add path="^/paynow\\.html$" \
\... dst-host="^www\\.example\\.com$"
[admin@MikroTik] ip hotspot walled-garden> print
Flags: X - disabled
 #   DST-HOST                    DST-PORT PATH                          ACTION
 0   ^www\.example\.com$                  ^/paynow\.html$               allow
[admin@MikroTik] ip hotspot walled-garden>
```

Notes:

- \\ symbol sequence is used to enter \ character
- **\.** 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

Customizing HotSpot Servlet

Description

Servlet Pages

The HotSpot servlet recognizes 5 different request types:

1. request for a remote host
   - if user is logged in, the requested page is served
   - if user is not logged in, but the destination host is allowed by walled garden, then the request is also served
   - if user is not logged in, and the destination host is disallowed by walled garden,

rlogin.html is displayed; if **rlogin.html** is not found, **redirect.html** is used to redirect to the login page

2. request for '/' on the HotSpot host
   - if user is logged in, **rstatus.html** is displayed; if **rstatus.html** is not found, **redirect.html** is used to redirect to the status page
   - if user is not logged in, **rlogin.html** is displayed; if **rlogin.html** is not found, **redirect.html** is used to redirect to the login page
3. request for '/login' page
   - if user has successfully logged in (or is already logged in), **alogin.html** is displayed; if **alogin.html** is not found, **redirect.html** is used to redirect to the originally requested page or the status page (in case, original destination page was not given)
   - if user is not logged in (username was not supplied, no error message appeared), **login.html** is showed
   - if login procedure has failed (error message is supplied), **flogin.html** is displayed; if **flogin.html** is not found, **login.html** is used
   - in case of fatal errors, **error.html** is showed
4. request for '/status' page
   - if user is logged in, **status.html** is displayed
   - if user is not logged in, **fstatus.html** is displayed; if **fstatus.html** is not found, **redirect.html** is used to redirect to the login page
5. request for '/logout' page
   - if user is logged in, **logout.html** is displayed
   - if user is not logged in, **flogout.html** is displayed; if **flogout.html** is not found, **redirect.html** is used to redirect to the login page

Note that if it is not possible to meet a request using the pages stored on the router's FTP server, the default pages are used.

There are many possibilities to customize what the HotSpot authentication pages look like:

- The pages are easily modifiable. They are stored on the router's FTP server in **hotspot** directory.
- By changing the variables, which client sends to the HotSpot servlet, it is possible to reduce keyword count to one (username or password; for example, the client's MAC address may be used as the other value) or even to zero (License Agreement; some predefined values general for all users or client's MAC address may be used as username and password)
- Registration may occur on a different server (for example, on a server that is able to charge Credit Cards). Client's MAC address may be passed to it, so that this information need not be written in manually. After the registration, the server may change RADIUS database enabling client to log in for some amount of time.

To insert variable in some place in HTML file, variable name surrounded by % symbols is used.

This construction may be used in any HotSpot HTML file accessed as '/', '/login', '/status' or '/logout'. For example, to show a link to the login page, following construction can be used:

```
<a href="%link-login%">login</a>
```

Variables

All of the Servlet HTML pages use variables to show user specific values. Variable names appear only in the source - they are automatically replaced with the respective values by the HotSpot Servlet. For each variable there is an example included in brackets.

- Common variables (available in all pages):
  - **hostname** - DNS name or IP address (if DNS name is not given) of the HotSpot Servlet ("hotspot.example.net")
  - **identity** - RouterOS identity name ("MikroTik")
  - **ip** - IP address of the client ("10.5.50.2")
  - **link-logout** - link to logout page ("http://10.5.50.1/logout")
  - **link-login** - link to login page including original URL requested ("http://10.5.50.1/login?dst=http://www.example.com/")
  - **link-status** - link to status page ("http://10.5.50.1/status")
  - **link-orig** - original URL requested ("http://www.example.com/")
  - **session-id** - value of 'session-id' parameter in the last request
  - **var** - value of 'var' parameter in the last request
- redirect.html, rlogin.html, rstatus.html, fstatus.html, flogout.html:
  - **link-redirect** - page to which redirect has to be done (for example, "http://www.example.com/")
- login.html, flogin.html:
  - **mac** - MAC address ("01:23:45:67:89:AB"; if unknown, then contains "---")
  - **error** - error message, if previous login failed ("invalid username or password")
  - **input-user** - name and value of username input field ("name=user value=john")
  - **input-password** - name of password input field ("name=password")
  - **input-popup** - name and value of pop-up input field ("name=popup checked")
  - **form-input** - name of input form and login JavaScript for password encoding ("name=login onSubmit=...")
  - **main** - MD5 encryption JavaScript and form for encrypted password
  - **user** - value of username input field ("john")
  - **domain** - value of domain ("example")
  - **popup** - whether to pop-up checkbox ("true" or "false")
  - **chap-id** - value of chap ID ("\371")
  - **chap-challenge** - value of chap challenge ("\357\015\330\013\021\234\145\245\303\253\142\246\133\175\375\316")
- alogin.html:

- **link-redirect** - page to which redirect has to be done ("http://www.example.com/")
- **login-time** - time in seconds after which redirect has to be done ("9")
- **popup** - if alogin.html should pop-up status page in new window ("true" or "false")

- logout.html:
  - **username** - name ("john")
  - **ip** - IP address ("192.168.0.222")
  - **mac** - MAC address ("01:23:45:67:89:AB")
  - **uptime** - session uptime ("10h2m33s")
  - **session-timeout** - session timeout left for the user ("5h" or "---" if none)
  - **session-valid-till** - date and time when session will expire ("Sep/21/2003 16:12:33" or "---" if there is no session-timeout)
  - **idle-timeout** - idle timeout ("20m" or "---" if none)
  - **bytes-in** - number of bytes received from the user ("15423")
  - **bytes-out** - number of bytes sent to the user ("11352")
  - **packets-in** - number of packets received from the user ("251")
  - **packets-out** - number of packets sent to the user ("211")
  - **uptime-secs** - uptime in seconds ("125")
  - **session-timeout-secs** - session timeout in seconds ("3475" or "" if there is such timeout)
  - **idle-timeout-secs** - idle timeout in seconds ("88" or "" if there is such timeout)
  - **limit-bytes-in** - byte limit for send ("1000000" or "---" if there is no limit)
  - **limit-bytes-out** - byte limit for receive ("1000000" or "---" if there is no limit)
  - **remain-bytes-in** - remaining bytes until limit-bytes-in will be reached ("337465" or "---" if there is no limit)
  - **remain-bytes-out** - remaining bytes until limit-bytes-out will be reached ("124455" or "---" if there is no limit)

- status.html:
  - **username** - name ("john")
  - **ip** - IP address ("192.168.0.222")
  - **mac** - MAC address ("01:23:45:67:89:AB")
  - **uptime** - session uptime ("10h2m33s")
  - **session-timeout** - session timeout left for the user ("5h" or "---" if none)
  - **session-valid-till** - date and time when session will expire ("Sep/21/2003 16:12:33" or "---" if there is no session-timeout)
  - **idle-timeout** - idle timeout ("20m" or "---" if none)
  - **bytes-in** - number of bytes received from the user ("15423")
  - **bytes-out** - number of bytes sent to the user ("11352")
  - **packets-in** - number of packets received from the user ("251")
  - **packets-out** - number of packets sent to the user ("211")
  - **refresh-time** - time in seconds after which to automatically refresh status page ("90s")

- **refresh-time-str** - more friendly representation of refresh-time ("1m30s")
- **uptime-secs** - uptime in seconds ("125")
- **session-timeout-secs** - session timeout in seconds ("3475" or "" if there is such timeout)
- **idle-timeout-secs** - idle timeout in seconds ("88" or "" if there is such timeout)
- **limit-bytes-in** - byte limit for send ("1000000" or "---" if there is no limit)
- **limit-bytes-out** - byte limit for receive ("1000000" or "---" if there is no limit)
- **remain-bytes-in** - remaining bytes until limit-bytes-in will be reached ("337465" or "---" if there is no limit)
- **remain-bytes-out** - remaining bytes until limit-bytes-out will be reached ("124455" or "---" if there is no limit)

- error.html:
  - **error** - error message ("DHCP lease not found")

Notes

To insert % symbol as a text (not as a part of variable construction), "%%" has to be used (if there is only one % symbol on a page or string between it and next % symbol is not a valid variable name, % may be used with the same result).

In most cases it is required login page to use **main** variable. And it is strongly suggested to place it BEFORE **form-input** input form. Otherwise situation can happen, that user already has entered his username/password, but MD5 encryption JavaScript is not yet loaded. It may result in password being sent over network in plain text. And of course, that login will fail in this case, too (if **allow-unencrypted-password** property is not set to **yes**).

The resulting password to be sent to the HotSpot gateway is formed MD5-hashing the concatenation of the following: chap-id, the password of the user and chap-challenge (in the given order)

The gateway uses CHAP authentication in case client's browser is hashing his/her pasword (in other words, if the **main** variable has been initialized successfully before the form is being submitted). In case plain-text password has been sent, PAP authentication algorithm is used. So if you want to force PAP-only authentication, you must remove the **main** variable from the servlet (of course, you must also allow the gateway to accept unencrypted passwords, or otherwise no one would be able to login at all).

In case if variables are to be used in link directly, then they must be escaped accordingly. For example, in login page??link will not work as intended, if username will be "123&456=1 2". In this case instead of %user%, its escaped version must be used: %user-esc%: link. Now the same username will be converted to "123%26456%3D1+2", which is the valid representation of "123&456=1 2" in URL. This trick may be used with any variables, not only with %user%.

Example

With basic HTML language knowledge and the examples below it should be easy to implement the ideas described above.

- To provide predefined value as username, in login.html change:

```
<input type="text" %input-user%>
```

  to this line:

```
<input type="hidden" name="user" value="hsuser">
```

  (where **hsuser** is the username you are providing)

- To provide predefined value as password, in login.html change:

```
<input type="password" %input-password%>
```

  to this line:

```
<input type="hidden" name="password" value="hspass">
```

  (where **hspass** is the password you are providing)

- To send client's MAC address to a registration server in form of:

```
https://www.server.serv/register.html?mac=XX:XX:XX:XX:XX:XX
```

  change the Login button link in login.html to:

```
https://www.server.serv/register.html?mac=%mac%
```

  (you should correct the link to point to your server)

- To show a banner after user login, in alogin.html after

```
if ('%popup%' == 'true') newWindow();
```

  add the following line:

```
open('http://your.web.server/your-banner-page.html', 'my-banner-name','');
```

  (you should correct the link to point to the page you want to show)

- To choose different page shown after login, in login.html change:

```
<input type="hidden" name="dst" value="%link-orig%">
```

  to this line:

```
<input type="hidden" name="dst" value="http://your.web.server">
```

  (you should correct the link to point to your server)

An another example is making HotSpot to authenticate on a remote server (which may, for example, perform creditcard charging):

- Allow direct access to the external server in dst-nat and hotspot-temp firewall chain or, alternatively, either in mangle, or in walled-garden. Note: walled-garden is not compatible with HTTPS.
- Modify login page of the HotSpot servlet to redirect to the external authentication server. The external server should modify RADIUS database as needed
  Here is an example of such a login page to put on the HotSpot router (it is redirecting to https://auth.example.com/login.php, replace with the actual address of an external authentication server):

```
<html> <title>...</title> <body> <form name="redirect"
action="https://auth.example.com/login.php" method="post"> <input type="hidden"
name="mac" value="%mac%"> <input type="hidden" name="ip" value="%ip%"> <input
```

```
type="hidden" name="user" value="%user%"> <input type="hidden" name="link-login"
value="%link-login%"> <input type="hidden" name="link-orig" value="%link-orig%">
<input type="hidden" name="error" value="%error%"> </form> <script
language="JavaScript"> <!-- document.redirect.submit(); //--> </script> </body>
</html>
```

- The external server can log in a HotSpot client by redirecting it back to the original HotSpot servlet login page, specifying the correct username and password
  Here is an example of such a page (it is redirecting to https://hotspot.example.com/login, replace with the actual address of a HotSpot router; also, it is displaying www.mikrotik.com after successful login, replace with what needed):
  ```
  <html> <title>Hotspot login page</title> <body> <form name="login"
  action="https://hotspot.example.com/login" method="post"> <input type="text"
  name="user" value="demo"> <input type="password" name="password" value="none">
  <input type="hidden" name="domain" value=""> <input type="hidden" name="dst"
  value="http://www.mikrotik.com/"> <input type="submit" name="login" value="log in">
  </form> </body> </html>
  ```
- Hotspot will ask RADIUS server whether to allow the login or not. If not allowed, alogin.html page will be displayed (it can be modified to do anything!). If not allowed, flogin.html (or login.html) page will be displayed, which will redirect client back to the external authentication server.
- Note: as shown in these examples, HTTPS protocol and POST method can be used to secure communications.

Possible Error Messages

Description

There are two kinds of errors: fatal non-fatal. Fatal errors are shown on a separate HTML page called error.html. Non-fatal errors are basically indicating incorrect user actions and are shown on the login form.

General non-fatal errors:

- **You are not logged in** - trying to access the status page or log off while not logged in. Solution: log in
- **IP <your_ip_address> is already logged in** - trying to log in while somebody from this IP address has already been logged in. Solution: you should not log in twice
- **no chap** - trying to log in using MD5 hash, but HotSpot server does not know the challenge used for the hash (this is may happen if you use BACK buttons in browser). Solution: instruct browsers to reload (refresh) the login page
- **invalid username: this MAC address is not yours** - trying to log in using a MAC address username different from the actual user's MAC address. Solution: no - users with usernames that look like a MAC address may only log in from the MAC address specified as their user name
- **current license allows only <num> sessions** - Solution: try to log in later when there will be less concurrent user sessions, or buy an another license that allows more simultaneous sessions
- **hotspot service is shutting down** - RouterOS is currently being restarted or shut down.

Solution: wait until the service will be available again

General fatal errors:

- **unknown MAC address for <your_ip_address>** - trying to log in from a remote MAC network (i.e. there is a router between the client and the HotSpot gateway). Cause: if auth-requires-mac parameter is enabled, users can only log in from the same MAC network the HotSpot router belongs to. Solution: disable the auth-requires-mac parameter
- **can't get IP: no IP pool** - DHCP-pool login method is chosen for this user, but no IP pool is specified. Solution: make sure that an IP pool is specified in /ip hotspot server submenu
- **no address from ip pool** - unable to get an IP address from an IP pool. Solution: make sure there is a sufficient amount of free IP addresses in IP pool
- **IP <your_ip_address> from pool is already logged in** - somebody is already logged in using the address should be given by DHCP server (in DHCP-pool login method) to the current user. Solution: do not specify static IP addresses from the range that belongs to an IP pool that HotSpot is using to dynamically give out IP addresses
- **unable to determine IP address of the client** - The client's IP address is the same the HotSpot router has. Cause: this happen if a user is using a local SOCKS proxy server to access the HotSpot gateway. Solution: do not use local SOCKS proxy to access the HotSpot page. You may use a local HTTP proxy server without any troubles
- **invalid license** - report this error to MikroTik
- **unencrypted passwords are not accepted** - received an unencrypted password. Solution: either use a browser that supports JavaScript (all modern browsers) or set allow-unencrypted-passwords parameter to yes

Local HotSpot user database non-fatal errors:

- **invalid username or password** - self-explanatory
- **invalid mac address** - trying to log in from a MAC address different from specified in user database. Solution: log in from the correct MAC address or take out the limitation
- **your uptime limit is reached** - self-explanatory
- **your traffic limit is reached** - either limit-bytes-in or limit-bytes-out limit is reached
- **no more sessions are allowed for user** - the shared-users limit for the user's profile is reached. Solution: wait until someone with this username logs out, use different login name or extend the shared-users limit

RADIUS client non-fatal errors:

- **invalid username or password** - RADIUS server has rejected the username and password sent to it without specifying a reason. Cause: either wrong username and/or password, or other error. Solution: should be clarified in RADIUS server's log files
- **<error_message_sent_by_radius_server>** - this may be any message (any text string) sent back by RADIUS server. Consult with your RADIUS server's documentation for further information

RADIUS client fatal errors:

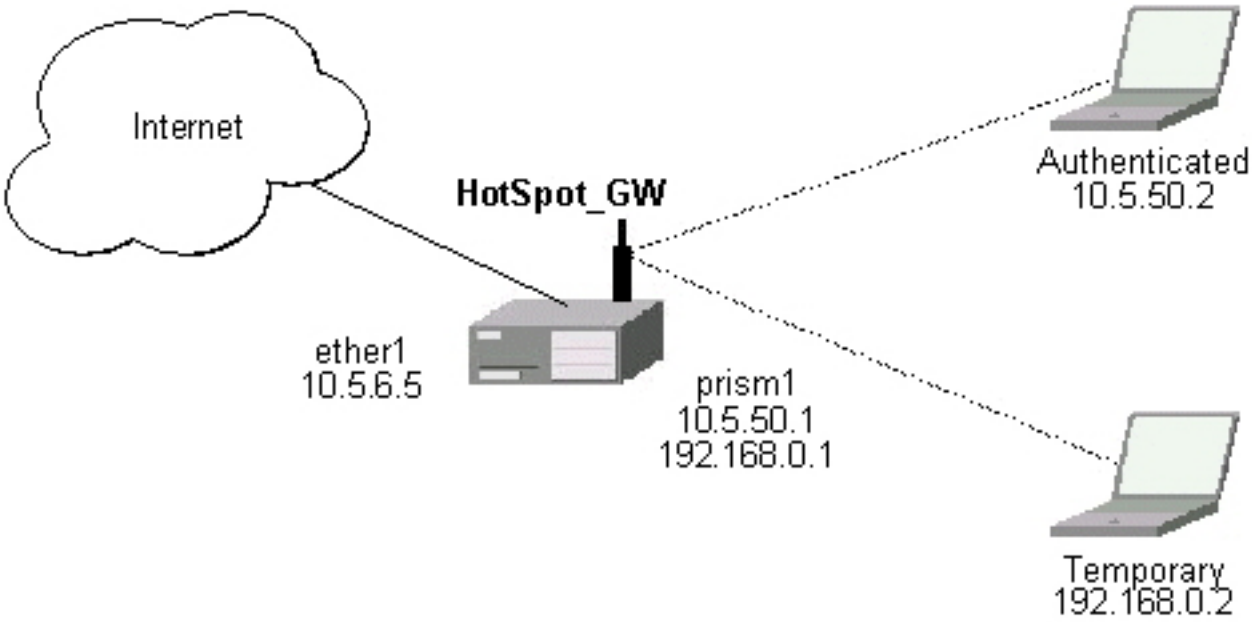
- **RADIUS server is not responding** - self-explanatory. Solution: check whether the RADIUS server is running and is reachable from the HotSpot router
- **invalid response from RADIUS server** - the RADIUS server has sent incorrect response

(neither accept nor reject). Solution: make sure the RADIUS server sends only accept or reject responses to authentication requests

Application Examples

Description

Let us consider following example HotSpot setup:



There will be 2 HotSpot IP address ranges used for clients on **prism1** interface. You are free to choose the address ranges, just make sure you use masquerading for not routed ones. In this example, we are using:

- temporary addresses which must be masqueraded:
  - network: 192.168.0.0/24
  - gateway: 192.168.0.1
  - pool: 192.168.0.2-192.168.0.254
- real addresses which require routing:
  - network: 10.5.50.0/24
  - gateway: 10.5.50.1
  - pool: 10.5.50.2-10.5.50.254

For HotSpot client accounting, HotSpot will add dynamic firewall rules in firewall HotSpot chain. This chain has to be created manually. And all network packets (to/from HotSpot clients) have to pass this chain.

Example

1. The **ether1** interface is configured with IP address 10.5.6.5/24 and the default route pointing to the 10.5.6.1 gateway.
2. The **prism1** interface is configured for AP mode and is able register IEEE 802.11b wireless clients. See the Prism Interface Manual for more details.
3. ARP should be set to **reply-only** mode on the **prism1** interface, so no dynamic entries are added to the ARP table. DHCP server will add entries only for clients which have obtained DHCP leases:

```
/interface prism set prism1 arp=reply-only
```

4. Add two IP addresses to the **prism1** interface:

```
/ip address add address=192.168.0.1/24 interface=prism1 \
    comment="hotspot temporary network"
/ip address add address=10.5.50.1/24 interface=prism1 \
    comment="hotspot real network"
```

5. add 2 IP address pools:

```
/ip pool add name=hs-pool-temp ranges=192.168.0.2-192.168.0.254
/ip pool add name=hs-pool-real ranges=10.5.50.2-10.5.50.254
```

6. add masquerading rule for temporary IP pool, which is not routed:

```
/ip firewall src-nat add src-address=192.168.0.0/24 action=masquerade \
    comment="masquerade hotspot temporary network"
```

   Make sure you have routing for authenticated address space. Try to ping 10.5.50.1 from your Internet gateway 10.5.6.1, for example. See the Basic Setup Guide on how to set up routing.

7. Add dhcp server (for temporary IP addresses):

```
/ip dhcp-server add name="hs-dhcp-server" interface=prism1 lease-time=14s \
    address-pool=hs-pool-temp add-arp=yes disabled=no
/ip dhcp-server network add address=192.168.0.0/24 gateway=192.168.0.1 \
    dns-server=159.148.60.2,159.148.108.1 domain="example.com"
```

8. Add hotspot server setup (for real IP addresses):

```
/ip hotspot server add name=hs-server dhcp-server=hs-dhcp-server \
    address-pool=hs-pool-real
/ip dhcp-server network add address=10.5.50.0/24 gateway=10.5.50.1 \
    dns-server=159.148.60.2,159.148.108.1 domain="example.com"
```

9. Add local hotspot user:

```
/ip hotspot user add name=Ex password=Ex
```

10. Setup hotspot service to run on port 80 (www service has to be assigned another port, e.g., 8081):

```
/ip service set www port=8081
/ip service set hotspot port=80
```

Note! Changing www service to other port than 80 requires that you specify the new port when connecting to MikroTik router using WinBox, e.g., use 10.5.50.1:8081 in this case.

11. Redirect all TCP requests from temporary IP addresses to hotspot service:

```
/ip firewall dst-nat add src-address=192.168.0.0/24 dst-port=443 protocol=tcp \
    action=redirect to-dst-port=443\
    comment="redirect unauthorized hotspot clients to hotspot service"
/ip firewall dst-nat add src-address=192.168.0.0/24 protocol=tcp \
    action=redirect to-dst-port=80 \
    comment="redirect unauthorized hotspot clients to hotspot service"
```

12. Allow DNS requests and ICMP ping from temporary addresses and reject everything else:

```
/ip firewall add name=hotspot-temp comment="limit unauthorized hotspot clients"

/ip firewall rule forward add src-address=192.168.0.0/24 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule input add src-address=192.168.0.0/24 dst-port=80 \
protocol=tcp action=accept comment="accept requests for hotspot servlet"
/ip firewall rule input add src-address=192.168.0.0/24 dst-port=443 \
protocol=tcp action=accept comment="accept request for hotspot servlet"
/ip firewall rule input add src-address=192.168.0.0/24 dst-port=67 \
protocol=udp action=accept comment="accept requests for local DHCP server"
/ip firewall rule input add src-address=192.168.0.0/24 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule hotspot-temp add protocol=icmp action=return \
    comment="allow ping requests"
/ip firewall rule hotspot-temp add protocol=udp dst-port=53 action=return \
    comment="allow dns requests"
/ip firewall rule hotspot-temp add action=reject \
    comment="reject access for unauthorized hotspot clients"
```

13. Add hotspot chain:

```
/ip firewall add name=hotspot comment="account authorized hotspot clients"
```

14. Pass all through-going traffic to the hotspot chain:

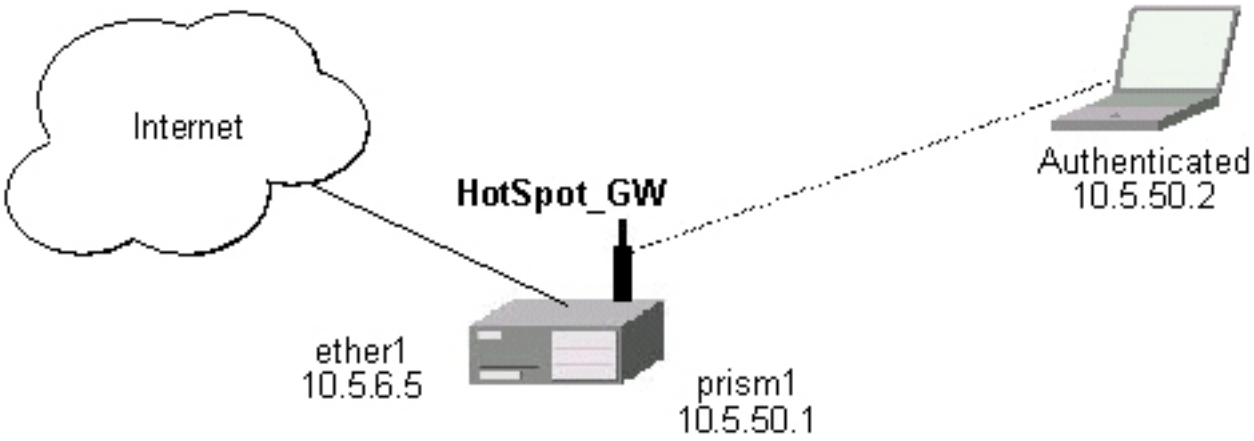
```
/ip firewall rule forward add action=jump jump-target=hotspot \
    comment="account traffic for authorized hotspot clients"
```

Note that in order to use SSL authentication, you should install an SSL certificate. This topic is not covered by this manual section. Please see the respective manual section on how to install certificates in MikroTik RouterOS

Application Examples

Description

Let us consider following example HotSpot setup:



There are clients at **prism1** interface, which are able to use Internet already. You want all these clients to authenticate before they are able to use Internet.

For hotspot client accounting, hotspot will add dynamic firewall rules in firewall hotspot chain. This chain has to be created manually. And all network packets (to/from hotspot clients) have to pass this chain.

Example

1. Setup hotspot service to run on port 80 (www service has to be assigned another port, e.g., 8081):

```
/ip service set www port=8081
/ip service set hotspot port=80
```

   Note! Changing www service to other port than 80 requires that you specify the new port when connecting to MikroTik router using WinBox, e.g., use 10.5.50.1:8081 in this case.

2. Setup hotspot profile to mark authenticated users with flow name "hs-auth":

```
/ip hotspot profile set default mark-flow="hs-auth" login-method=enabled-address
```

3. Add local hotspot user:

```
/ip hotspot user add name=Ex password=Ex
```

4. Redirect all TCP requests from unauthorized clients to the hotspot service:

```
/ip firewall dst-nat add in-interface="prism1" flow="!hs-auth" protocol=tcp \
    dst-port=443 action=redirect to-dst-port=443 \
    comment="redirect unauthorized hotspot clients to hotspot service"
/ip firewall dst-nat add in-interface="prism1" flow="!hs-auth" protocol=tcp \
    action=redirect to-dst-port=80 \
    comment="redirect unauthorized clients to hotspot service"
```

5. Allow DNS requests and ICMP ping from temporary addresses and reject everything else:

```
/ip firewall add name=hotspot-temp comment="limit unauthorized hotspot clients"
```

```
/ip firewall rule forward add in-interface=prism1 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule input add in-interface=prism1 dst-port=80 protocol=tcp \
action=accept comment="accept requests for hotspot servlet"
/ip firewall rule input add in-interface=prism1 dst-port=443 protocol=tcp \
action=accept comment="accept request for hotspot servlet"
/ip firewall rule input add in-interface=prism1 dst-port=67 protocol=udp \
protocol=udp action=accept comment="accept requests for local DHCP server"
/ip firewall rule input add in-interface=prism1 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized hotspot clients"

/ip firewall rule hotspot-temp add flow="hs-auth" action=return \
    comment="return if connection is authorized"
/ip firewall rule hotspot-temp add protocol=icmp action=return \
    comment="allow ping requests"
/ip firewall rule hotspot-temp add protocol=udp dst-port=53 action=return \
    comment="allow dns requests"
/ip firewall rule hotspot-temp add action=reject \
    comment="reject access for unauthorized clients"
```

6. Create a hotspot chain for authorized hotspot clients:

```
/ip firewall add name=hotspot comment="account authorized hotspot clients"
```

7. Pass all through-going traffic to the hotspot chain:

```
/ip firewall rule forward add action=jump jump-target=hotspot \
    comment="account traffic for authorized hotspot clients"
```

Note that in order to use SSL authentication, you should install an SSL certificate. This topic is not covered by this manual section. Please see the respective manual section on how to install certificates in MikroTik RouterOS

As we see from example, only hotspot interface is used - we don't care what IP addresses are there.

It is possible to add hotspot authentication for one more interface (**prism2**) by adding only 4 additional firewall rules:

- Setup dst-nat to redirect unauthorized clients to the hotspot service:

```
/ip firewall dst-nat add in-interface="prism2" flow="!hs-auth" protocol=tcp \
    dst-potr=443 action=redirect to-dst-port=443 \
    comment="redirect unauthorized prism2 clients to hotspot service"
/ip firewall dst-nat add in-interface="prism2" flow="!hs-auth" protocol=tcp \
    action=redirect to-dst-port=80 \
    comment="redirect unauthorized prism2 clients to hotspot service"
```

- Limit access for unauthorized **prism2** interface clients:

```
/ip firewall rule forward add in-interface=prism2 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized prism2 clients"
/ip firewall rule input add in-interface=prism2 action=jump \
jump-target=hotspot-temp comment="limit access for unauthorized prism2 clients"
```

Optional Settings

- You may want to use same address space for both your LAN and HotSpot networks. Please consult the IP Address and ARP Manual for **proxy-arp** feature.
- You may want to translate the destination addresses of all TCP port 25 connections (SMTP) from HotSpot users to your local mail sever for mail relaying. Thus, users can retain their mail client setup and use your mail server for outgoing mail without reconfiguring their mail clients. If **10.5.6.100** is your mail server accepting connections from network **10.5.50.0/24**, then the required destination NAT rule would be:

```
/ip firewall dst-nat add src-address=10.5.50.0/24 dst-port=25 protocol=tcp \
    to-dst-address=10.5.6.100 action=nat \
    comment="Translate SMTP TCP 25 port to our mail server"
```

- One more option is to allow access certain pages without authentication (walled garden). For example, if **http://hotspot.example.com** is your web server's name:

```
[admin@MikroTik] ip hotspot walled-garden> add \
\... dst-host="^hotspot\\.example\\.com$"
[admin@MikroTik] ip hotspot walled-garden> print
Flags: X - disabled
 #   DST-HOST                    DST-PORT PATH                          ACTION
 0   ^hotspot\.example\.com$                                            allow
[admin@MikroTik] ip hotspot walled-garden>
```

- For HotSpot clients to use transparent web-proxy on the same router, following configuration can be used:
  1. make sure, **web-proxy** software package is installed and DNS client is configured
  2. it is assumed, that HotSpot is set up and successfully running on port 8088. Hotspot clients are connected to the interface named **prism1**
  3. set up HotSpot to use one of the router's local IP addresses (10.5.50.1):

```
/ip hotspot set hotspot-address=10.5.50.1
```

  4. set up web-proxy to run on the same IP address on the port 3128:

```
/ip web-proxy set enabled=yes src-address=10.5.50.1:3128 transparent-proxy=yes
```

  5. configure hotspot service to use this web proxy as its parent proxy:

```
/ip hotspot set parent-proxy=10.5.50.1:3128
```

  6. redirect all requests from hotspot interface to port 80 (except to 10.5.50.1), to the web-proxy:

```
/ip firewall dst-nat add in-interface=prism1 dst-address=!10.5.50.1/32 \
    dst-port=80 protocol=tcp action=redirect
    to-dst-port=8088 comment="transparent proxy"
```

  7. Now, everything should be working fine. Only traffic of the redirected requests to the web-proxy will not be accounted. It's because this traffic will not pass through the

forward chain.
to enable accounting for the HotSpot user traffic to/from transparent web-proxy, additional firewall rules should be added:

```
/ip firewall rule input add in-interface=prism1 dst-port=3128 \
    protocol=tcp action=jump jump-target=hotspot \
    comment="account traffic from hotspot client to local web-proxy"
/ip firewall rule output add src-port=3128 protocol=tcp \
    out-interface=prism1 action=jump jump-target=hotspot \
    comment="account traffic from local web-proxy to hotspot client"
```

- You may want to allow multiple logins using the same username/password. Set the argument value of **shared-users** to the number of simultaneous user sessions using the same username in HotSpot profile. For example, to allow 10 clients to use the same username simultaneously:

```
/ip hotspot profile set default shared-users=10
```

- If you want the router to resolve DNS requests, enable DNS cache, and redirect all the DNS requests to the router itself (**159.148.60.2** is this example mean the external DNS server the router will work with):

```
/ip dns set primary-dns=159.148.60.2
/ip dns set allow-remote-requests=yes
/ip firewall dst-nat add protocol=udp dst-port=53 action=redirect \
comment="intercept all DNS requests"
```