Router User AAA

Document revision 2.4 (February 6, 2008, 1:40 GMT)
This document applies to V3.0

Table of Contents

General Information

Summary

This documents provides summary, configuration reference and examples on router user management.

Specifications

Packages required: ***system***
License required: ***level1***
Home menu level: ***/user***
Hardware usage: ***Not significant***

Description

MikroTik RouterOS router user facility manage the users connecting the router from the local console, via serial terminal, telnet, SSH or Winbox. The users are authenticated using either local database or designated RADIUS server.

Each user is assigned to a user group, which denotes the rights of this user. A group policy is a combination of individual policy items.

In case the user authentication is performed using RADIUS, the RADIUS client should be previously configured under the **/radius** submenu.

Router User Groups

Home menu level: ***/user group***

Description

The router user groups provide a convenient way to assign different permissions and access rights to different user classes.

Property Description

name (*name*) - the name of the user group

policy (*multiple choice: local | telnet | ssh | ftp | reboot | read | write | policy | test | winbox | password | web | sniff*) - group policy item set

- **local** - policy that grants rights to log in locally via local console
- **telnet** - policy that grants rights to log in remotely via telnet
- **ssh** - policy that grants rights to log in remotely via secure shell protocol
- **ftp** - policy that grants remote rights to log in remotely via FTP and to transfer files from and to the router. Keep in mind that the user allowed to transfer files, may also upload a new RouterOS version that will be applied upon the next reboot
- **reboot** - policy that allows rebooting the router
- **read** - policy that grants read access to the router's configuration. All console commands that do not alter router's configuration are allowed
- **write** - policy that grants write access to the router's configuration, except for user management. This policy does not allow to read the configuration, so make sure to enable read policy as well
- **policy** - policy that grants user management rights. Should be used together with write policy
- **test** - policy that grants rights to run ping, traceroute, bandwidth-test and wireless scan, sniffer and snooper commands
- **winbox** - policy that grants rights to connect to the router remotely using WinBox interface
- **password** - policy that grants user option to change own password
- **web** - policy that grants rights to log in remotely via WebBox
- **sniff** - policy that grants access to the packet sniffer facility

Notes

There are three system groups which cannot be deleted:

```
[admin@rb13] > /user group print
 0 name="read" policy=local,telnet,ssh,reboot,read,test,winbox,password,web,
                     sniff,!ftp,!write,!policy

 1 name="write" policy=local,telnet,ssh,reboot,read,write,test,winbox,password,
                      web,sniff,!ftp,!policy

 2 name="full" policy=local,telnet,ssh,ftp,reboot,read,write,policy,test,winbox,
                     password,web,sniff
[admin@rb13] >
```

Exclamation sign '**!**' just before policy item name means **NOT**.

Example

To add **reboot** group that is allowed to reboot the router locally or using telnet, as well as read the router's configuration, enter the following command:

```
[admin@rb13] user group> add name=reboot policy=telnet,reboot,read,local
[admin@rb13] user group> print
 0 name="read" policy=local,telnet,ssh,reboot,read,test,winbox,password,web,
                     sniff,!ftp,!write,!policy

 1 name="write" policy=local,telnet,ssh,reboot,read,write,test,winbox,password,
                      web,sniff,!ftp,!policy

 2 name="full" policy=local,telnet,ssh,ftp,reboot,read,write,policy,test,winbox,
                     password,web,sniff
 3 name="reboot" policy=local,telnet,reboot,read,!ssh,!ftp,!write,!policy,!test,
                       !winbox,!password,!web,!sniff
[admin@rb13] user group>
```

Router Users

Home menu level: ***/user***

Description

Router user database stores the information such as username, password, allowed access addresses and group about router management personnel.

Property Description

address (*IP addressnetmask*; default: **0.0.0.0/0**) - host or network address from which the user is allowed to log in

group (*name*) - name of the group the user belongs to

name (*name*) - user name. Although it must start with an alphanumeric character, it may contain "*", "_", "." and "@" symbols

password (*text*; default: "") - user password. If not specified, it is left blank (hit [Enter] when logging in). It conforms to standard Unix characteristics of passwords and may contain letters, digits, "*" and "_" symbols

Notes

There is one predefined user with full access rights:

```
[admin@MikroTik] user> print
Flags: X - disabled
  #   NAME                                            GROUP ADDRESS
  0   ;;; system default user
      admin                                           full  0.0.0.0/0

[admin@MikroTik] user>
```

There always should be at least one user with full access rights. If the user with full access rights is the only one, it cannot be removed.

Example

To add user **joe** with password **j1o2e3** belonging to **write** group, enter the following command:

```
[admin@MikroTik] user> add name=joe password=j1o2e3 group=write
[admin@MikroTik] user> print
Flags: X - disabled
  0   ;;; system default user
      name="admin" group=full address=0.0.0.0/0

  1   name="joe" group=write address=0.0.0.0/0


[admin@MikroTik] user>
```

Monitoring Active Router Users

Command name: ***/user active print***

Description

This command shows the currently active users along with respective statisics information.

Property Description

address (*read-only: IP address*) - host IP address from which the user is accessing the router
- **0.0.0.0** - the user is logged in locally from the console

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

radius (*read-only: flag*) - the user has been authenticated through a RADIUS server

via (*read-only: console | telnet | ssh | winbox*) - user's access method
- **console** - user is logged in locally
- **telnet** - user is logged in remotely via telnet
- **ssh** - user is logged in remotely via secure shell protocol
- **winbox** - user is logged in remotely via WinBox tool

when (*read-only: date*) - log in date and time

Example

To print currently active users, enter the following command:

```
[admin@rb13] user> active print
Flags: R - radius
 #   WHEN                NAME                                            ADDRESS
VIA
 0   feb/27/2004 00:41:41 admin                                           1.1.1.200
ssh
 1   feb/27/2004 01:22:34 admin                                           1.1.1.200
winbox
[admin@rb13] user>
```

Router User Remote AAA

Home menu level: ***/user aaa***

Description

Router user remote AAA enables router user authentication and accounting via RADIUS server.

Property Description

accounting (yes | no; default: **yes**) - whether to use RADIUS accounting

default-group (*name*; default: **read**) - user group used for the users authenticated via a RADIUS server by default (if the server did not specify a different user group)

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

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

Notes

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

Example

To enable RADIUS AAA, enter the following command:

```
[admin@MikroTik] user aaa> set use-radius=yes
[admin@MikroTik] user aaa> print
        use-radius: yes
        accounting: yes
    interim-update: 0s
     default-group: read
[admin@MikroTik] user aaa>
```

SSH keys

Home menu level: ***/user ssh-keys***

Description

Remote users may be allowed to log in without using password authentication and even ever entering their password, but by using pregenerated DSA openssh SSH keys instead. Note that if you use puttygen, convert generated keys to right type.

Property Description

key-owner (*read-only: text*) - emote user, as specified in the key file

user (*name*) - the user that is allowed to log in using this key (must exist in the user list)

Command Description

import - import the uploaded DSA key

- **user** - the user the imported key is linked to
- **file** - filename of the DSA key to import

Example

Generating key on a linux machine:

```
sh-3.00$ ssh-keygen -t dsa -f ./id_dsa
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_dsa.
Your public key has been saved in ./id_dsa.pub.
The key fingerprint is:
91:d7:08:be:b6:a1:67:5e:81:02:cb:4d:47:d6:a0:3b admin-ssh@test
```

Importing the generated (ang uploaded) key:

```
[admin@MikroTik] user ssh-keys> print
 # USER                 KEY-OWNER
[admin@MikroTik] user ssh-keys> import file=id_dsa.pub user=admin-ssh
[admin@MikroTik] user ssh-keys> print
 # USER                 KEY-OWNER
 0 admin-ssh            admin-ssh@test
[admin@MikroTik] user ssh-keys>
```