RADIUS MANAGER 3

INSTALLATION MANUAL

Version 3.8

10/25/2010

TABLE OF CONTENTS

FOREWORD

This document describes the installation procedure of Radius Manager billing system on a Linux host.

The document covers the installation steps of Radius Manager billing system on two major Linux distributions:

1. **Redhat** based systems: Fedora Core 5-12, CentOS 5+, RHEL 5+
2. **Debian** based systems: Debian 4+, Ubuntu 8+

For beginners we recommend the usage of Fedora Core 8 or newer versions. Fedora Core is the easiest and the most comfortable Linux system available nowadays. It comes with all required packages to install and run Radius Manager. The packages are available on the installation media and they are also downloadable from the official online repositories using the Yum tool.

If You are using a different type of Linux, please read this manual carefully and substitute the paths and filenames with those ones which are available in your system.

In this document You can also find guidelines how to set up your NAS (Network Access Server) to use with Radius Manager system.

Radius Manager currently supports the following NAS types:

- **Mikrotik 2.8+** Use final releases only, the usage of RC (release candidate) versions are not recommended. Supported main features are: PPPoE, PPtP, L2tP, Hotspot, SSID RADIUS MAC authentication.
- **ChilliSpot 1.1** running on Linux. You can download a tested version from our download portal.
- **StarOS v2 or v3** server. Supported features are: full PPPoE and limited RADIUS access list support.
- **Cisco VPDN server** (PPP) with the appropriate IOS version. VPDN and Virtual template support are necessary.
- **pfSense** Hotspot server.

To successfully install Radius Manager on your host, You have to complete the following steps:

1. Install ionCube runtime libraries
2. Build and configure FreeRadius server
3. Configure MySQL database and credentials
4. Install Radius Manager WEB components
5. Install Radius Manager binaries
6. Complete the post installation steps and fine tuning

With the help of this installation manual You can set up Radius Manager billing system on your host successfully. If You have problems during the installation, please contact the customer support on the following email address: support@dmasoftlab.com

INSTALLATION

Prerequisites

To successfully install and run Radius Manager, You need the following components installed on the Linux host:

Required hardware and software components:

1. **x86 compatible CPU** (32 or 64 bit, single or multiple core)
2. **FreeRadius 2.1.8 DMA Softlab mod 2** (downloadable from www.dmasoftlab.com)
3. **PHP 5** or better
4. **MySql 5** or better
5. **MySQL development libraries**
6. **php-mysql**
7. **php-mcrypt**
8. **curl, php-curl**
9. **glibc 2.3** or better
10. **libstdc++ 5**
11. **C/C++** compiler
12. **ionCube** runtime libraries. The libraries are reely available on www.ioncube.com and on www.dmasoftlab.com
13. **Javascript** enabled browser on running on client machines

Optional components:

1. **Webmin** for easy administration of databases and Linux system (www.webmin.com)
2. **phpMyAdmin** for database maintenance (www.dmasoftlab.com)

Linux system preparation

Fedora

Install the necessary components on your Linux host before You begin the installation of Radius Manager.

1. Disable SeLinux in */etc/sysconfig/selinux* and reboot your host:

```
SELINUX=disabled
```

2. Install **MySQL development** libraries, **curl**, **php-mysql**, **php-mcrypt**, **libstdc++ 5** and **libtool-ltdl**:

```
[root@localhost]# yum install mysql-devel
[root@localhost]# yum install curl
[root@localhost]# yum install php-mysql
[root@localhost]# yum install php-mcrypt
[root@localhost]# yum install compat-libstdc++-33
[root@localhost]# yum install libtool-ltdl-devel
```

On Fedora Core 11 and 12 do not install libtool-ltdl-devel. Delete it if it is already installed:

```
[root@localhost]# rpm –e libtool-ltdl-devel
```

Debian, Ubuntu

If You are planning to use Radius Manager on Debian, Ubuntu systems, You have to install **MySQL development** libraries, **perl** subsystem, **curl**, **php5-curl**, **php5-mysql**, **php5-mcrypt**, **php5-cli**, **libstdc++ 5** and **libtool-ltdl-devel**:

```
[root@localhost]# apt-get install libmysqlclient15-dev
[root@localhost]# apt-get install libperl-dev
[root@localhost]# apt-get install curl
[root@localhost]# apt-get install php5-cli
[root@localhost]# apt-get install php5-mysql
[root@localhost]# apt-get install php5-mcrypt
[root@localhost]# apt-get install php5-curl
[root@localhost]# apt-get install libstdc++5
[root@localhost]# apt-get install libltdl3-dev
```

If apt-get cannot download and install libtool 1.x or libstdc++ 5, You can download them manually from http://www.dmasoftlab.com/downloads.

The downloaded files can be installed with **dpkg** command:

```
[root@localhost]# wget http://www.dmasoftlab.com/cont/download/libltdl3_1.5.24-
1ubuntu1_i386.deb
[root@localhost]# dpkg -i libltdl3_1.5.24-1ubuntu1_i386.deb
```

The same applies for libstdc++ 5.

Installation procedure of ionCube runtime system

Radius Manager requires ionCube runtime libraries. You can download them from:

http://www.dmasoftlab.com/downloads

Before installing ionCube, You have to know the following:

1. The Linux system architecture (32 or 64 bit)
2. Which **PHP version** are You using
3. Where is your **php.ini** file located

Example ionCube installation

1. **Copy** and untar the **ionCube runtime libraries** (32 or 64 bit – use the correct archive) to */usr/local/ioncube*. Use Midnight Commander or other tool to do this.

2. Add the appropriate ionCube loader to your *php.ini*. For example, on a Linux running PHP 5.2.2 You will have to add the following line:

```
zend_extension=/usr/local/ioncube/ioncube_loader_lin_5.2.so
```

Be sure to use the correct PHP version in the zend_extension line.

If there are other zend_extension entries in the php.ini file, place this new entry before the existing entries.

Please note. on Debian based systems there are two php.ini files:

/etc/php5/apache2/php.ini
/etc/php5/cli/php.ini

You have to add the ionCube loaders in **both files**! On Fedora systems there is only one *php.ini* file available: */etc/php.ini*

3. Test the ionCube loaders from shell:

```
[root@localhost]# php -v
PHP 5.1.2 (cli) (built: Feb 28 2006 06:21:15)
Copyright (c) 1997-2006 The PHP Group
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies
    with the ionCube PHP Loader v3.1.31, Copyright (c) 2002-2007, by ionCube Ltd.
```

You have to see the ionCube PHP Loader version displayed correctly.

4. **Restart** the web server (Fedora):

```
[root@localhost]# sevice httpd restart
```

On Debian:

```
[root@localhost]# apache2ctl restart
```

5. Run **ifconfig** command from shell to determine the MAC address of the network interface card (NIC):

```
[root@localhost]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:E8:EC:8A:E8
          inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:e8ff:feec:8ae8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13287 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3683486 (3.5 MiB)  TX bytes:6942105 (6.6 MiB)
          Interrupt:10 Base address:0xd800
```

6. Now it's time to request a trial license. Log on to DMA Softlab customer's portal (https://customers.dmasoftlab.com) and request a **trial license** for the **hardware address** (MAC address) of your network interface card.

   Radius Manager will run only on the specified host and the license is binding to the MAC address of the network interface card. You can migrate Radius Manager to another host if You also move the same network interface card with it.

   It is strongly recommended to request a license for a **removable networking interface** to allow migration to new host without loosing the license.

7. When a license file is issued (You will get a notification about it in email), download and copy the *lic.txt* and *mod.txt* to **radiusmanager** web directory (read the "Installation procedure of Radius Manager" chapter of this manual) to enable licensing of your Radius Manager installation.

Troubleshooting the ionCube loader system

If encoded files fail to run, you can test ionCube runtime by using the helper PHP script *ioncube-loader-helper.php*, which is included in the loader download archive.

1. **Copy** the *ioncube-encoded-file.php* PHP script to your **http root** (on Redhat-based system it is */var/www/html*).
2. **Access** the *ioncube-encoded-file.php* script using your favorite web browser: http://yourhost/ioncube-encoded-file.php
3. If You can see the message "*This file has been successfully decoded. ionCube Loaders are correctly installed*", it means You have successfully installed ionCube runtime on your host and it is ready to use. If You can't decode the file via a HTTP call, check the php.ini and be sure **SeLinux is disabled**.

Notes about PHP safe mode

If PHP safe mode is enabled in *php.ini*, it can prevent the execution of UNIX commands called via **shell_exec** from Radius Manager if the additional parameters are not configured properly. We recommend to turn off the PHP safe mode feature to enable all functionalities of Radius Manager. Please always check the Apache log files if You encounter any PHP / Apache related problems.

Installation procedure of FreeRadius

Follow the installation steps to successfully build, install and configure FreeRadius RADIUS server on your host. Use only FreeRadius 2.1.8 DMA Softlab mod source archive (downloadable from our site). It is compiled by our team and it is 100% compatible with Radius Manager.

Other versions and builds will not function properly with Radius Manager. If your host already has a different version of FreeRadius installed, remove it completely including the configuration files (/etc/raddb or /usr/local/etc/raddb).

Execute the following actions as super user (root user):

1. **Download FreeRadius** archive from the following URL:

   http://www.dmasoftlab.com/downloads

2. Build **FreeRadius** server from sources. Do it in the following way.

   Ungzip and **untar** the FreeRadius archive:

```
[root@localhost]# gzip -d freeradius-server-2.1.8-dmamod-2.tar.gz
[root@localhost]# tar xvf freeradius-server-2.1.8-dmamod-2.tar
```

   Create the makefile:

```
[root@localhost]# cd freeradius-server-2.1.8
[root@localhost]# ./configure
```

   On some 64 bit systems it is necessary to specify the MySQL library path:

```
[root@localhost]# ./configure --with-mysql-lib-dir=/usr/lib64/mysql
```

   Build and **install** the system:

```
[root@localhost]# make
[root@localhost]# make install
```

   Be sure You have the **mysql-devel** package installed. By default, FreeRadius will be installed in */usr/local* directory.

3. Now You can **test** FreeRadius in debug mode. Start it with parameter -X:

```
[root@localhost]# radiusd -X
...
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on command file /usr/local/var/run/radiusd/radiusd.sock
Listening on proxy address * port 1814
Ready to process requests.
```

It must answer with "*Ready to process requests*". If *radiusd* cannot find the required libraries, issue *ldconfig* from shell to refresh the ld linker's cache.

```
[root@localhost]# ldconfig
```

If there are still problems, please contact the customer support using the following email address: support@dmasoftlab.com.

4. **Set** the correct **permissions** on FreeRadius configuration files (Fedora):

```
[root@localhost]# chown apache /usr/local/etc/raddb
[root@localhost]# chown apache /usr/local/etc/raddb/clients.conf
```

Debian:

```
[root@localhost]# chown www-data /usr/local/etc/raddb
[root@localhost]# chown www-data /usr/local/etc/raddb/clients.conf
```

Radius Manager updates the clients.conf automatically, so it is necessary to set the correct permission on it. **Do not modify** the clients.conf by hand. Don't forget to define all NASes in ACP with the correct secret and restart FreeRadius (from ACP or from shell) after modifying the NASes in the system.

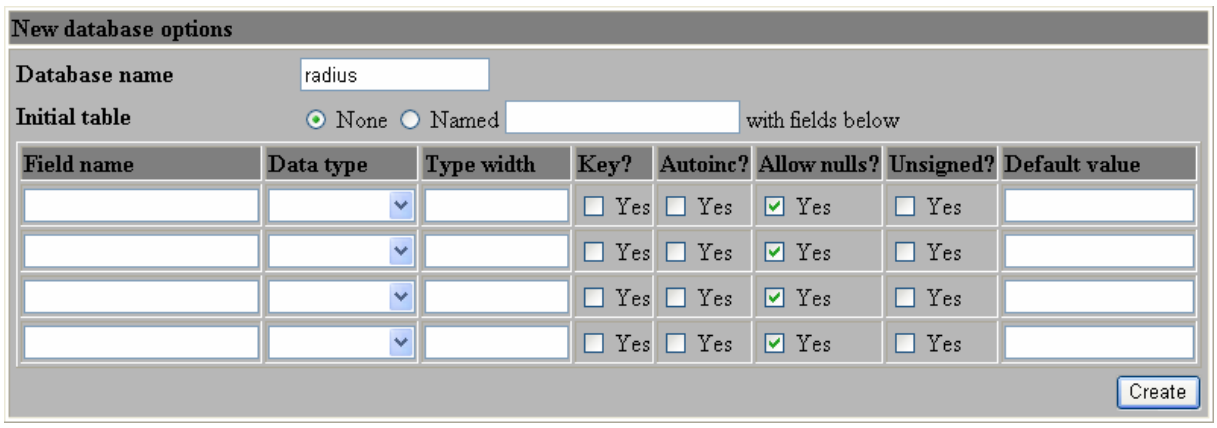
5. **Review** and modify (if needed) the **MySQL credentials** in */usr/local/etc/raddb/sql.conf*:

```
# Connection info:
server = "localhost"
#port = 3306
login = "radius"
password = "radius123"
```

6. Create **MySQL databases**, **credential**. Two methods are described here: **MySQL** command line tool and **Webmin**.

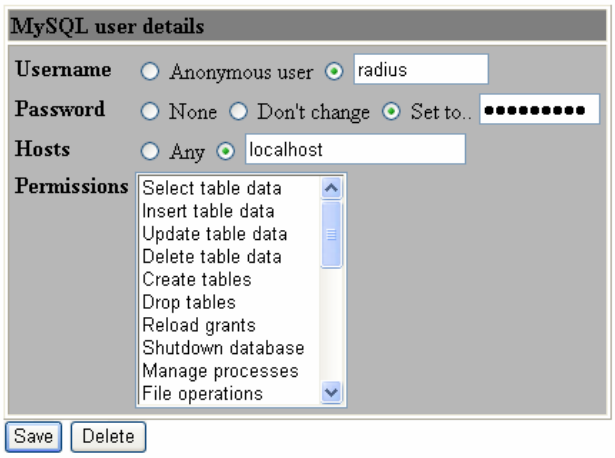
Creating databases with Webmin

Webmin is ideal for beginners. Create the **RADIUS** and **CONNTRACK** databases with it:

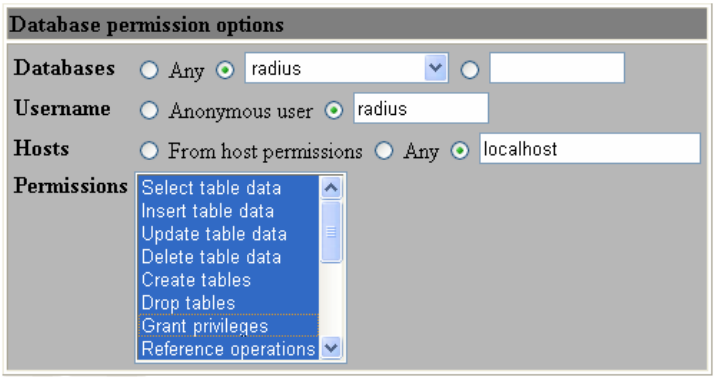


Define the database name in the proper field (first create RADIUS then CONNTRACK).

Create **database users**. For initial installation use password **radius123** for user **radius,** and **conn123** for user **conntrack**.



Don't forget to define the **host permissions**. Select all permissions for both **radius** and **conntrack** users.



Creating databases with MySQL command line tool

If You are familiar with MySql command line tool, You can create databases, users and permissions with it easily and much faster than Webmin method.

Log on to MySql server as root:

```
[root@localhost]# mysql -u root –ppassword
```

Where *password* is the MySql root password. If there is no password for root, simply invoke MySql program with command **mysql**.

Execute the following statement group from the MySQL command shell:

```
CREATE DATABASE radius;
CREATE DATABASE conntrack;
CREATE USER 'radius'@'localhost' IDENTIFIED BY 'radius123';
CREATE USER 'conntrack'@'localhost' IDENTIFIED BY 'conn123';
GRANT ALL ON radius.* TO radius@localhost;
GRANT ALL ON conntrack.* TO conntrack@localhost;
```

Completing this step the databases are ready to use.

Installation procedure of Radius Manager

There are two methods of installation available:

1. **Interactive**, using the included installer script.
2. **Manual** installation, using Unix commands and / or Midnight Commander.

Interactive installation

The easiest way to install Radius Manager is to use the included *install.sh* script. It is located in Radius Manager tar archive and can be used on Redhat, Debian and (with slight modification of the environment) on other systems. Before You begin, be sure You have prepared the MySQL database tables and credentials. Radius Manager requires two databases:

1. **RADIUS** – for storing all system data, including users and accounting information.
2. **CONNTRACK** – for storing connection tracking system (CTS) data. Create both databases even on a non-CTS enabled system.

After You decompress the Radius Manager tarball (use command *tar xf [filename]*), invoke the installer script, but first change its permission to 755. In the examples below we will use the installer script on Redhat / Fedora system.

```
[root@localhost]# chmod 755 install.sh
[root@localhost]# ./install.sh
Radius Manager installer
Copyright 2004-2010, DMA Softlab LLC
All right reserved.

(Use CTRL+C to abort any time)

Select the type of your operating system:
1. Redhat (Fedora, CentOS etc.)
2. Debian (Ubuntu etc.)

Choose an option: [1]
```

Select the operating system You have. For Redhat, RHEL, CentOS, Fedora select option **1**. If You have a Debian or Ubuntu select **2**.

Now select the installation method:

```
Select installation type:
1. New installation
2. Upgrade old system

Choose an option: [1]
```

For new installation, use option **1**. You can see the default options after every question, so You can just press enter in most cases.

```
Choose an option: [1] 1
Selected installation method: NEW INSTALLATION
WWW root path: [/var/www/html]
```

Now define the **HTTP root folder**. The installer will create *radiusmanager* subfolder in it automatically. On Redhat You can simply press enter.

Now define the MySQL database credentials:

```
RADIUS database host: [localhost]
RADIUS database username: [radius]
RADIUS database password: [radius123]
CTS database host: [localhost]
CTS database username: [conntrack]
CTS database password: [conn123]
```

For the default setup simply press enter and use MySQL user “radius” with password “radius123” for RADIUS database, and conntrack / conn123 for CONNTRACK database. The host is “localhost” by default. If You have different setup, specify proper values. If You are planning to use the system with hundreds of online users, it is recommended to use separate database host for CONNTRACK database.

In the next step You have to define the FreeRadius user. It must be the correct user to set the permission properly on /etc/*radiusmanager.cfg*. If there are permission problems on /etc/*radiusmanager.cfg*, Radius Manager binaries will not function at all.

```
Freeradius UNIX user: [root]
```

On Fedora it is **root**, so simply press enter.

Now define the HTTP user (the user name under Apache is running). It is required to set the permission on files in *radiusmanager/config* directory. On Fedora it is the **apache** user.

```
Httpd UNIX user: [apache]
```

You can now decide to create **rmpoller** service or not? It is a standard Fedora / Debian compatible service script which invokes rmpoller helper. You can also start rmpoller using alternative ways.

```
Create rmpoller service: [y]
```

In most cases simply press enter. When a service has been created, You can use the command (on Fedora)

service rmpoller [start | stop]

to control **rmpoller** service activity. Also make this service auto starting at boot time together with FreeRadius. Use command *chkconfig –add rmpoller on* or use Webmin to activate the service at boot time.

In the next step select yes if You want to create the **rmconntrack** service. It is a standard Linux service, like rmpoller. It is required for **Radius Manager CTS** only.

```
Create rmconntrack service: [y]
```

When a service has been created, You can use the command

service rmconntrack [start | stop]

to control **rmconntrack** service activity. Also make this service auto starting at boot time.

It is strongly recommended to create a full database backup before You continue. Answer '**yes**' to the following question:

```
Create database backup: [y]
```

Now the system warns You it will **overwrite** the existing databases if You continue. Press '**y**' to continue or '**n**' to abort the installation process.

```
WARNING! If You continue You will overwrite the existing RADIUS database!

Are You sure to start the installation? [n] y
```

You can press **Ctrl+C** any time to abort the installation process.

```
Starting installation process...

Backing up radiusmanager.cfg
Backing up system_cfg.php
Backing up netcash_cfg.php
Backing up paypal_cfg.php
Backing up authorizenet_cfg.php
Backing up dps_cfg.php
Copying web content to /var/www/html/radiusmanager
Copying binaries to /usr/local/bin
Copying rootexec to /usr/local/sbin
Copying radiusmanager.cfg to /etc
Creating database backup
Creating mysql tables
Creating rmpoller service
Creating rmconntrack service
Copying logrotate script
Setting permission on raddb files
Copying radiusd init script to /etc/init.d

Installation finished!
```

When the installation process is finished, You can begin configuring the system with /etc/*radiusmanager.cfg* and *radiusmanager/config* files.

Add the following line to */etc/crontab* to execute *rmscheduler.php* every day after midnight:

```
02 0 * * * root /usr/bin/php /var/www/html/radiusmanager/rmscheduler.php [password]
```

Always define the full path of the PHP interpreter. If You are not sure, check it's location before You add the crontab record. The password has to match the predefined one in *system_cfg.php*. By default, the password is 12345, which must match the password defined in *system_cfg.php*.

Install the license files (*lic.txt* and *mod.txt*) in radiusmanager web folder and try to access the ACP (Administration Control Panel). Reboot your system to check if helper services are starting properly (radiusd, rmpoller and optionally rmconntrack).

To test RADIUS communication, be sure MySQL server is running. Start FreeRadius in debug mode:

```
[root@localhost]# radiusd -X
...
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on command file /usr/local/var/run/radiusd/radiusd.sock
Listening on proxy address * port 1814
Ready to process requests.
```

On the second terminal issue the **radtest** command:

```
[root@localhost]# radtest user 1111 localhost 1812 testing123
Sending Access-Request of id 57 to 127.0.0.1 port 1812
        User-Name = "user"
        User-Password = "1111"
        NAS-IP-Address = 127.0.0.1
        NAS-Port = 1812
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=57, length=50
        WISPr-Bandwidth-Max-Up = 262144
        WISPr-Bandwidth-Max-Down = 262144
        Acct-Interim-Interval = 60
```

You have to see **Access-Accept** answer. If You see an error message, check the following:

- Is MySQL server running?
- Are MySQL credentials correct?
- Are MySQL table permissions correct?
- Can FreeRadius connect to MySQL database?
- Have You created the RADIUS and CONNTRACK databases and tables?
- Is the NAS defined in ACP? In this case it is 127.0.0.1 ?(NAS-IP-Address = 127.0.0.1).
- If the hostname is different than localhost, You have to substitute the localhost with the IP address of the Linux server. You have to update the NAS list in RM ACP in this case.

To determine the correct NAS IP address, do the following:

- Stop radius daemon:

```
[root@localhost]# service radiusd stop
```

or

```
[root@localhost]# ps ax | grep radius
[root@localhost]# kill [pid]                    (Substitute the PID with the given PID)
```

Invoke debug mode:

```
[root@localhost]# radiusd -X
```

Try **radtest** now or try to authenticate users. In the debug output You will see the correct NAS-IP-Address which You have to enter in Radius Manager ACP / NAS editing form.

Manual installation

1. Copy **rmauth, rmacnt, rmpoller** and **rmconntrack** binaries with *cp* command or with Midnight Commander in */usr/local/bin* folder

2. **Set 755 permission** on all files:

```
[root@localhost]# chmod 755 /usr/local/bin/rmauth
[root@localhost]# chmod 755 /usr/local/bin/rmacnt
[root@localhost]# chmod 755 /usr/local/bin/rmpoller
[root@localhost]# chmod 755 /usr/local/bin/rmconntrack
```

3. Copy **radiusmanager.cfg** in */etc* folder.

4. **Edit the parameters** in *radiusmanager.cfg* to fit your needs.

5. **Change permission** on *radiusmanager.cfg* to ensure only FreeRadius user can access it:

```
[root@localhost]# chmod 600 /etc/radiusmanager.cfg
[root@localhost]# chown root.root /etc/radiusmanager.cfg
```

You have to chown this file to the correct user. It must be the user under FreeRadius is running, otherwise the binaries cannot read the configuration file and the authentication will fail.

6. Test **rmauth** from shell:

```
[root@localhost]# rmauth -v
rmauth version 3.8.0, build 1174 (20100329)
Copyright 2004-2010, DMA Softlab LLC
All rights reserved.
```

You have to see similar output to this. If there are errors it means You have an older glibc installed or some other libraries are missing from your Linux installation. In this case please contact the customer support (support@dmasoftlab.com) for the proper version of binaries or upgrade your system fit the requirements.

Test the database connectivity:

```
[root@localhost]# rmauth 192.168.0.8 user 1
Mikrotik-Xmit-Limit=1028,Mikrotik-Rate-Limit="262144/262144"
```

You have to see similar output to this. If there is a MySQL socket error, define the correct socket location in */etc/radiusmanager.cfg*. The default socket file on Redhat is **/var/lib/mysql/mysql.sock**. On Debian systems the proper socket path is **/var/run/mysqld/mysqld.sock**.

To successfully test *rmauth*, You have to create NAS entries in ACP. In this example, the NAS IP 192.168.0.8 was already defined in Radius Manager ACP and set as Mikrotik. You have to restart FreeRadius every time when You modify the NAS devices. Unfortunately FreeRadius doesn't read the configuration files dynamically.

7. Copy **rootexec** to */usr/local/sbin* folder.

8. Change **permission** on *rootexec* to 4755:

```
[root@localhost]# chmod 4755 /usr/local/sbin/rootexec
```

Rootexec is required to execute external UNIX commands from Radius Manager WEB interface. For security purposes it uses a password. The password prevents executions of binaries by anyone who has PHP script on the accounting server.

9. Create a crontab (*/etc/crontab*) entry for *rmscheduler.php*:

```
02 0 * * * root /usr/bin/php /var/www/html/radiusmanager/rmscheduler.php [password]
```

Always define the full path of the PHP interpreter. If You are not sure, check it's location first before You add the crontab record. The password has to match the defined one in the *system_cfg.php*. The default password is 12345.

10. **Copy** the complete Radius Manager web content to **http root** directory.

11. **Protect** the web configuration files (in config www folder) to be readable by **root** and **Apache** only (on Debian it is the **www-data** user):

```
[root@localhost]# chown apache system_cfg.php
[root@localhost]# chmod 600 system_cfg.php
[root@localhost]# chown apache paypal_cfg.php
[root@localhost]# chmod 600 paypal_cfg.php
[root@localhost]# chown apache netcash_cfg.php
[root@localhost]# chmod 600 netcash_cfg.php
[root@localhost]# chown apache authorizenet_cfg.php
[root@localhost]# chmod 600 authorizenet_cfg.php
```

12. **Edit the definitions** in *system_cfg.php* and optionally in *paypal_cfg.php, netcash_cfg* and *authorizenet_cfg.php.* Read the Reference chapter in this manual for details.

13. **Install** initial database **tables.** Use **MySQL** command line tool to do this:

```
[root@localhost]# mysql –u radius –pradius123 radius < radius.sql
[root@localhost]# mysql –u conntrack  –pconn123 conntrack < conntrack.sql
```

14. Start your web browser and check the functionality of the **Administration Control Panel** (ACP):

    http://yourhost/radiusmanager/admin.php

Use the following username and password:

Username: **admin**
Password: **1111**

Log in and try to access various functions. The initial manager name is **admin**.

Also test the functionality of the **User Control Panel** (UCP):

http://yourhost/radiusmanager/user.php

The initial username and password are:

Username: **user**
Password: **1111**

To be able to log on to UCP as another user, create the user in ACP first.

Note

By default, many web servers can list the contents of the directory where Radius Manager files are stored in. To prevent this there are several methods available:

- Use *.htaccess file* (for Apache, use the **Options -Indexes** directive; example file is included in *radiusmanager* folder). Be sure to enable .htaccess support for Apache (use *AllowOverride All* directive).
- Or edit the *httpd.conf* to disable completely the directory listing (remove the **Indexes** directive of the appropriate directory).

UPGRADING PROCEDURE

There are two upgrade methods available:

1. Interactive
2. Manual

Both methods require manual installation and configuration of FreeRadius server. This task is described here first.

Upgrading FreeRadius

The current version of Radius Manager system requires FreeRadius 2.1.8 DMA Sofltab mod 2. Install it if it is not yet installed on your host.

Read the appropriate chapter of this installation manual how to install FreeRadius server. Before You begin to install the new version of FreeRadius, **rename** the **raddb** folder to **raddb.bak**. With this You allow FreeRadius to install the new configuration files. Without this step the configuration files will remain unchanged and FreeRadius will not function properly with the old, incompatible configuration entries.

Configure FreeRadius with files in raddb folder as it is described in FreeRadius installation chapter.

Do not forget to set the proper permission on raddb files.

After that select the installation method for Radius Manager and continue reading the appropriate chapter.

Interactive upgrade

The Radius Manager installer script also supports interactive upgrade of the existing system. To do this, stop the running Radius Manager daemon processes (Redhat):

```
[root@localhost]# service rmpoller stop
[root@localhost]# service rmconntrack stop
```

On other systems use the following method (it can also be used on Redhat, too). Be sure to enter the proper PID for the **kill** command.

```
[root@localhost]# ps ax | grep rm
10205 ?        Ssl    0:25 /usr/local/bin/rmpoller
15917 ?        Ssl    5:08 /usr/local/bin/rmconntrack
[root@localhost]# kill 10205
[root@localhost]# kill 15917
```

Now You can untar the Radius Manager tar archive. CD to its folder and invoke the script *install.sh*. If the script *install.sh* isn't executable, change its permission to 755:

```
[root@localhost]# chmod 755 install.sh
[root@localhost]# ./install.sh
Radius Manager installer
Copyright 2004-2010, DMA Softlab LLC
All right reserved.

(Use CTRL+C to abort any time)

Select installation type:

1. New installation
2. Upgrade old system
3. Exit

Choose an option: [1] 2
```

For upgrading your current system, use option **2**. After selecting the upgrade mode, You have to choose the **currently installed** version.

WARNING! Be sure to select the correct installed version, otherwise the database gets corrupted!

```
Selected installation method: UPGRADE

0.  v1.1.5
1.  v2.0.0
2.  v2.0.1
3.  v2.0.2
4.  v2.5.0
5.  v2.5.1
6.  v3.0.0
7.  v3.0.1
8.  v3.1.0
9.  v3.1.1
10. v3.1.2
11. v3.2.0
12. v3.2.1
13. v3.2.2
14. v3.3.0
15. v3.4.0
16. v3.4.1
17. v3.5.0
18. v3.6.0
19. v3.6.1
20. v3.7.0

Select current installed version: 5
```

After defining the currently installed version of Radius manager, You have to enter the location of **http root** folder (webroot):

```
Current installed version is 2.5.1
WWW root path: [/var/www/html]
Directory /var/www/html/radiusmanager already exists. Overwrite? [n] y
```

It will ask to allow the overwriting of existing files in radiusmanager folder or not? Enter '**y**' to this question. The installer will backup the configuration files in *config* folder, so You can migrate the existing configuration later.

Now define the MySQL database access data:

```
RADIUS database host: [localhost]
RADIUS database username: [radius]
RADIUS database password: [radius123]
CTS database host: [localhost]
CTS database username: [conntrack]
CTS database password: [conn123]
```

Assuming the default setup simply press enter and use MySQL user "**radius**" with password "**radius123**" for RADIUS database, and "**conntrack**", "**conn123**" for CONNTRACK database. The host is "localhost" by default. If You have different setup, specify the correct data. When You use the system with hundreds of online users, it is recommended to use separate host for CONNTRACK database.

Define the FreeRadius user. It must be the correct user to set the permission on *radiusmanager.cfg.* If there are permission problems on *radiusmanager.cfg*, helper binaries will not work properly.

```
Freeradius UNIX user: [root]
```

On Fedora it is **root**, so simply press enter.

Now define the HTTP user (the username under Apache is running). On Fedora, it is the **apache** user, on Debian it is **www-data**. It is required to set the permissions on configuration files in *config* folder properly.

```
Httpd UNIX user: [apache]
```

You can now decide to create **rmpoller** service or not? It is a standard Linux service which invokes rmpoller helper. You can also start rmpoller in alternative ways.

```
Create rmpoller service: [y]
```

On Fedora simply press enter. When the service has been created, You can use the command

service rmpoller [start | stop]

to control the **rmpoller** service activity. Also make this service auto starting at boot time, together with FreeRadius.

Choose yes, if You want to create the **rmconntrack** service. It is a standard Fedora service like rmpoller. It is required for **Radius Manager CTS** only.

```
Create rmconntrack service: [y]
```

When a service has been created, You can use the command

service rmconntrack [start | stop]

to control the **rmconntrack** service activity. Also, make this service auto starting at boot time together with FreeRadius.

It is strongly recommended to create a full database backup before You continue. Answer '**y**' to the following question:

```
Create database backup: [y]
```

When all data were entered, the system will ask You to begin the upgrade procedure:

```
WARNING! Create a full database backup before You proceed!

Are You sure to start the upgrade? [n] y
```

Be sure You have created a **full database backup** before starting the upgrade procedure!

Press '**y**' to continue with the upgrade or '**n**' to abort the process.

You can use **Ctrl+C** any time to abort the installation process.

```
Starting installation process...

Backing up radiusmanager.cfg
Backing up paypal_cfg.php
Backing up system_cfg.php
Backing up netcash_cfg.php
Copying web content to /var/www/html/radiusmanager
Copying binaries to /usr/local/bin
Copying rootexec to /usr/local/sbin
Copying radiusmanager.cfg to /etc
Upgrading mysql tables. Please be patient.
Upgrading to version 3.0.0
Upgrading to version 3.0.1
Upgrading to version 3.1.0
Upgrading to version 3.1.1
Upgrading to version 3.1.2
Upgrading to version 3.2.0
Upgrading to version 3.2.1
Upgrading to version 3.2.2
Upgrading to version 3.2.3
Upgrading to version 3.3.0
Upgrading to version 3.4.0
Upgrading to version 3.4.1
Upgrading to version 3.5.0
Upgrading to version 3.5.1
Upgrading to version 3.6.0
Upgrading to version 3.6.1
Upgrading to version 3.7.0
Upgrading to version 3.8.0
Creating rmpoller service
Creating rmconntrack service

Configuration files are:
/var/www/html/radiusmanager/config/system_cfg.php
/var/www/html/radiusmanager/config/paypal_cfg.php
/var/www/html/radiusmanager/config/netcash_cfg.php
/var/www/html/radiusmanager/config/authorizenet_cfg.php
/etc/radiusmanager.cfg

Installation finished!
```

When the upgrade procedure is finished You have to see **no error** messages displayed. Now You can begin configuring the system with *radiusmanager.cfg*, *system_cfg.php, paypal_cfg.php, netcash_cfg.php* and *authorizenet_cfg.php* files.

Add the following line to */etc/crontab* to execute *rmscheduler* every day after midnight:

```
02 0 * * * root /usr/bin/php /var/www/html/radiusmanager/rmscheduler.php [password]
```

Always define the full path of the PHP interpreter. If You are not sure, check it's location first before You add the crontab record. The password has to match the defined one in the *system_cfg.php*. It is 12345 by default.

Manual upgrade

When You manually upgrade existing Radius Manager system, You have to check / reinstall / configure the following components:

1. Install ionCube runtime if not yet installed
2. Install the new version of FreeRadius if not yet installed
3. Install the new Radius Manager binaries
4. Upgrade RADIUS databases to the current version
5. Install new Radius Manager web files

Installing ionCube runtime

It is required to install ionCube runtime system if it is not installed on your host. Read about ionCube installation from chapter “Installation procedure of ionCube runtime system” of this manual.

Upgrading FreeRadius server

It is required to install FreeRadius 2.1.8 DMA Softlab mod 2 to use this release of Radius Manager billing system. Read about FreeRadius installation from chapter “Installation procedure of FreeRadius” of this manual.

Upgrading Radius Manager executables

Install the new **rmauth, rmacnt, rmpoller, rmconntrack** and **rootexec** executables. Follow **points 1 – 12** from chapter “Manual installation”. You have to stop the daemons before You can overwrite the old versions (rmpoller and rmconntrack). To do this, issue the following commands (Redhat):

```
[root@localhost]# service rmpoller stop
[root@localhost]# service rmconntrack stop
```

On other systems use the following method (it can also be used on Redhat). Be sure to enter the proper PID for **kill** command.

```
[root@localhost]# ps ax | grep rm
10205 ?        Ssl    0:25 /usr/local/bin/rmpoller
15917 ?        Ssl    5:08 /usr/local/bin/rmconntrack
[root@localhost]# kill 10205
[root@localhost]# kill 15917
```

Upgrading SQL tables

To upgrade from older Radius Manager to the newest, You have to **execute all SQL upgrade scripts** in correct order for both **RADIUS** and **CONNTRACK** database. For example if You are upgrading Radius Manager from 3.2.1 to 3.8.0, You have to execute the SQL scripts for RADIUS database in the following order:

1. upgrade-3.2.1_3.2.2.sql
2. upgrade-3.2.2_3.3.0.sql
3. upgrade-3.3.0_3.4.0.sql
4. upgrade-3.4.0_3.4.1.sql
5. upgrade-3.4.1_3.5.0.sql
6. upgrade-3.5.0_3.6.0.sql
7. upgrade-3.6.0_3.6.1.sql
8. upgrade-3.6.1_3.7.0.sql
9. upgrade-3.7.0_3.8.0.sql

For CONNTRACK database You have to execute the following scripts in correct order:

1. upgrade_cts-3.2.2_3.3.0.sql
2. upgrade_cts-3.3.0_3.4.0.sql
3. upgrade_cts-3.4.0_3.4.1.sql
4. upgrade_cts-3.4.1_3.5.0.sql
5. upgrade_cts-3.5.0_3.6.0.sql
6. upgrade_cts-3.6.0_3.6.1.sql
7. upgrade_cts-3.6.1_3.7.0.sql
8. upgrade_cts-3.7.0_3.8.0.sql

Please note the first CONNTRACK update script begins from 3.2.2. CTS system was introduced in version 3.2.2.

Check and update the service settings using the ACP after the system has been upgraded.

Installing new PHP files

Copy the new **radiusmanager** web directory, overwriting the old version. Be sure to backup the old *system_cfg.php, paypal_cfg.php, netcash_cfg.php* and *authorizenet_cfg.php* files before overwriting them. When it is done, review and modify the new configuration files in *config* directory. These files are changing from version to version, so You have to edit them every time after You have upgraded the system.

Add the following line to */etc/crontab* to execute *rmscheduler* every day after midnight:

```
02 0 * * * root /usr/bin/php /var/www/html/radiusmanager/rmscheduler.php [password]
```

Always define the full path of the PHP interpreter. If You are not sure, check its location before You add a crontab record. The password has to match the defined one in the *system_cfg.php*. The default password is 12345

WARNING!

- When upgrading to 3.0.0, invoice sum and payout data will be lost due to the new data storage mechanism.
- Always create **full database backup** before You begin the upgrade procedure!
- When upgrading to 3.8.0 the old invoice sums can be wrong. This is caused by a completely new organization of the rm_invoices table. If You have not printed the old invoices, please do it before You upgrade the system to v 3.8.0. Be sure You have created a full backup of database before You proceed the upgrade.

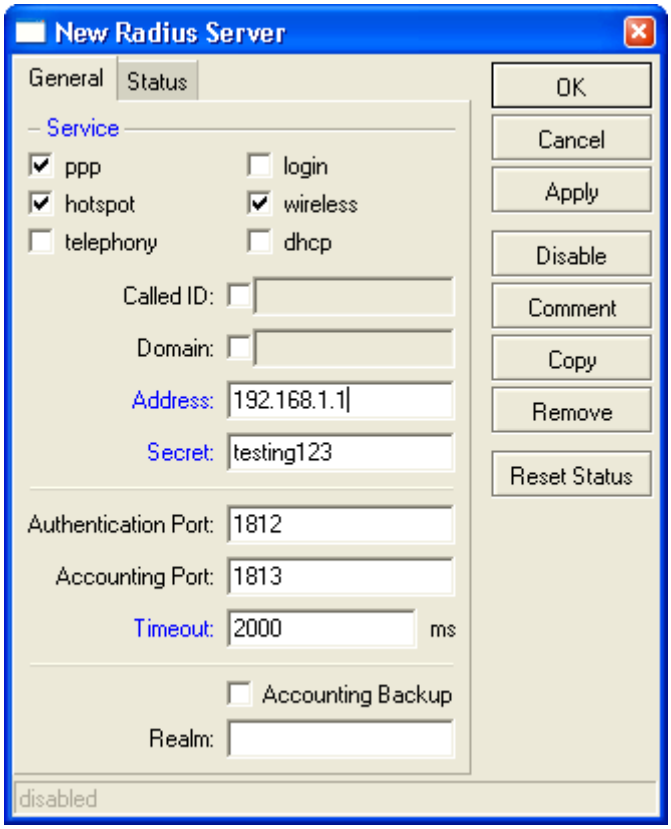
NAS CONFIGURATION

Mikrotik

Setting up RADIUS authentication and accounting

To send authentication and accounting requests to Radius server, You have to configure your Mikrotik NAS. Use Winbox to view and edit the configuration. Follow these steps:

1. **Connect** to your Mikrotik router using Winbox.
2. Select **Radius** from the main menu.
3. Click on the **+** to create a **new RADIUS** server description:

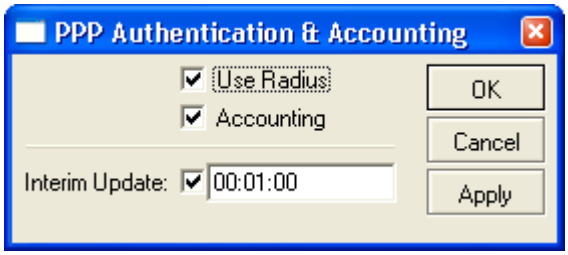


Description of fields:

- **Service**:
  - **Hotspot**: enable Hotpsot RADIUS authentication
  - **Wireless**: enable wireless access list RADIUS authentication (turn off Default authenticate for Hotspot wireless interface and turn on RADIUS MAC authentication for the WLAN interface)
  - **PPP**: for PPP RADIUS authentication
  - **Login**: Winbox (telnet, ssh) authentication from RADIUS
  - **Telephony**: telephony authentication from RADIUS
- **Address** is your RADIUS server host.
- **Secret** is the NAS secret from */usr/local/etc/raddb/clients.conf*
- **Authentication and Accounting** ports are the standard RADIUS ports.

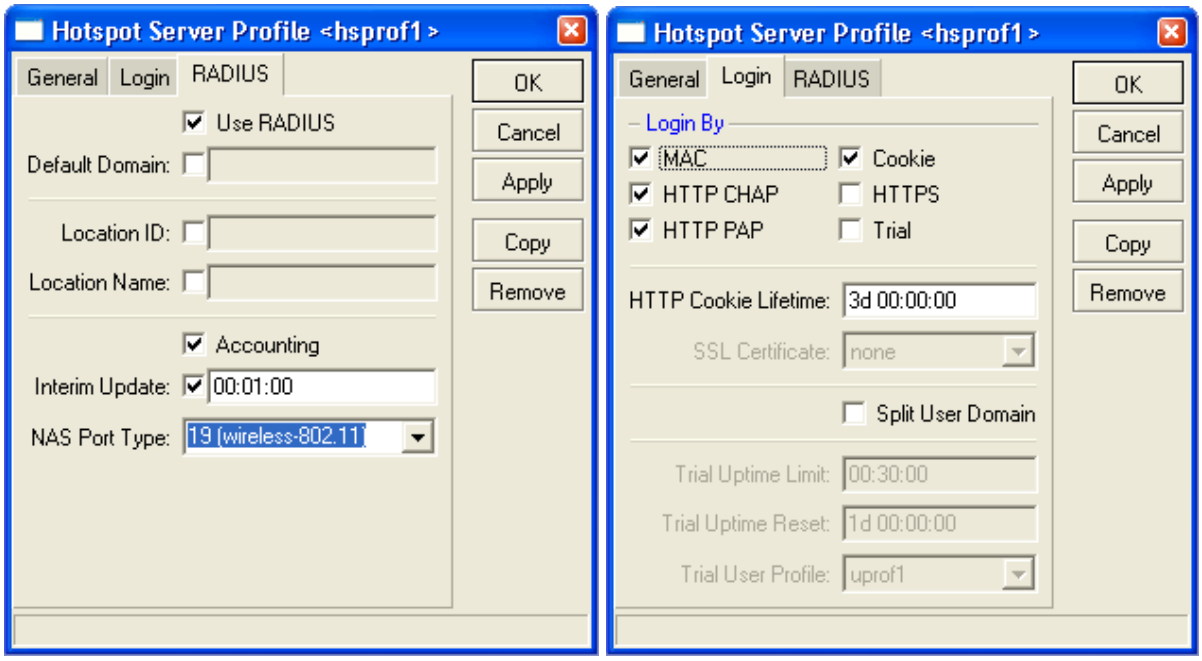
- **Timeout** defines how much milliseconds can elapse while the answer arrives from the RADIUS server. If You are using slower connection to RADIUS server or the accounting tables are large, set this timeout higher (3000-5000 ms).

4. **Set the AAA options** for PPP services (if You are using PPtP or PPPoE):



Turn on RADIUS authentication (**Use Radius**) and RADIUS accounting (**Accounting**). **Interim update** is the time interval when RADIUS client (Mikrotik NAS) sends the accounting information to the RADIUS server. If You have more than 200 online users, use higher values (5-8 minutes) to avoid MySQL overload.

5. Set the AAA options and authentication method for **Hotspot service**:



Hotspot server profile options:

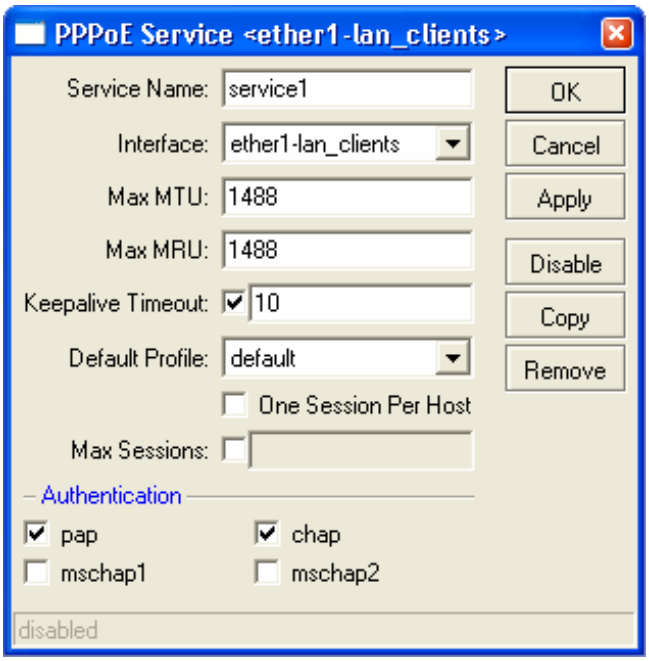
- **Use RADIUS** – this option is used to send access-request packets to RADIUS server.
- **Accounting** – this option is used to send the accounting data to the RADIUS server.
- **Interim update** – defines the interval when the RADIUS accounting data are periodically refreshed. Use a numeric value of 1-5 minutes here. Lower values generate heavy load on MySQL server.

Login By options:

- **MAC** – MAC-only authentication is used for Hotspot clients.
- **HTTP CHAP** – defines HTTP CHAP authentication method. It uses encrypted packets to send the username / password information from NAS to RADIUS server. Always use CHAP if your CPE devices support it.

- **HTTP PAP** – defines HTTP PAP authentication method; it is a non-encrypted method to send the username / password from NAS to RADIUS server.
- **Cookie** – Hotspot login page will remember the username / password entered.
- **HTTP cookie lifetime** – Defines how many days to remember the username / password.

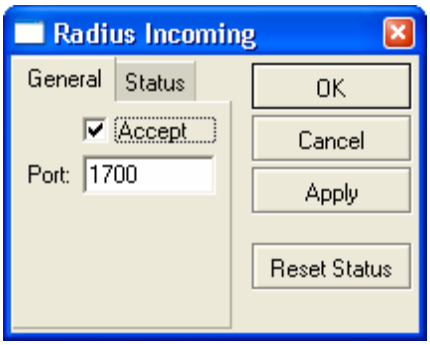
6. Set the AAA options and authentication method for **PPPoE service**:



You have to define the following data:

- **Service name –** it is a reference for PPPoE clients.
- **Interface** – The name of the **interface** where PPPoE server is listening.
- The max **MTU** and **MRU** values (use the default values or a bit smaller, for example 1480).
- **PAP** or **CHAP** authentication method (don't use MSCHAP1 or MSCHAP2).
- **Default profile** – Create a new profile and select it from this list.
- **Keepalive timeout** – Define 30-60 seconds here.

7. **Enable incoming Radius** requests (POD packets). It is requires to use remote disconnection method:

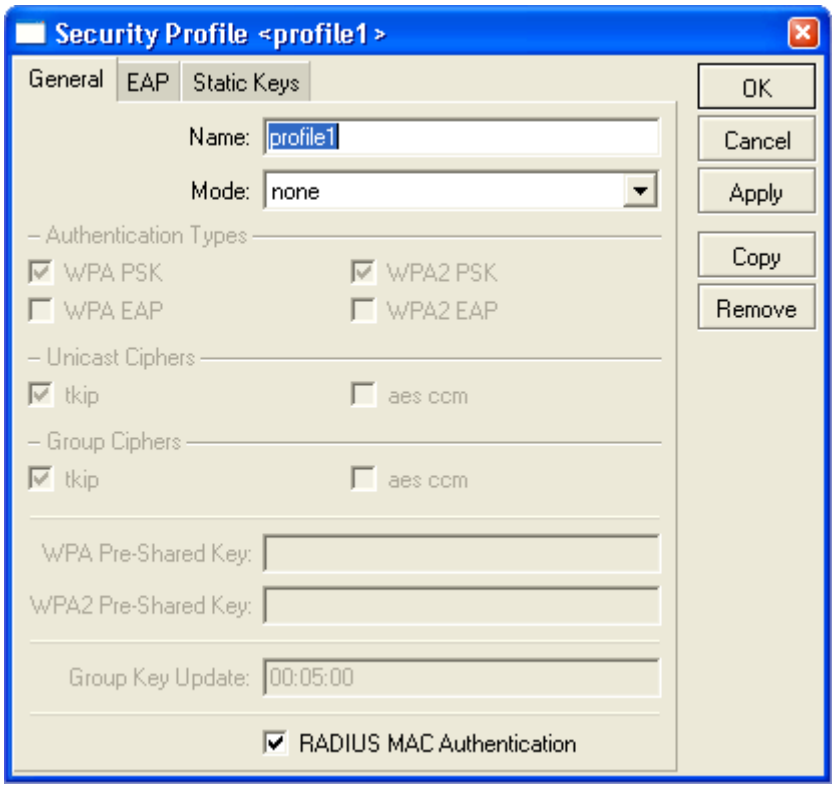


Don't forget to open the **UDP port 1700** in firewall on Mikrotik and Linux server.

Enabling RADIUS access list support (RADIUS ACL)

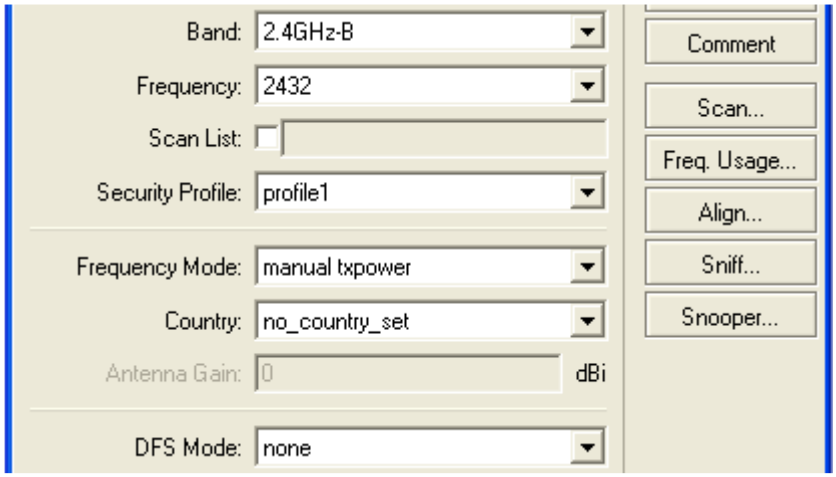
By default, all wireless clients can connect to your Mikrotik AP. If You want to filter them and allow only registered clients to connect to your SSID, You have enable RADIUS MAC authentication in Mikrotik AP.

1. **Create a security profile** using Winbox:



Set the checkbox for RADIUS MAC Authentication.

2. **Assign the security profile** to the wireless interface:



In this case when a client tries to connect to the SSID, Mikrotik authenticates the client's MAC address using the RADIUS server. If the MAC can be found in the database, Mikrotik allows the connection.

If You are planning to use Instant Access Services (IAS), install the customized **login.html** file which can be found in Radius Manager tar archive in *www/mikrotik* folder.

ChilliSpot

Radius Manager is fully compatible with Linux version of ChilliSpot 1.1.0 Hotspot server. The only one limitation is: You can't use more than one simultaneous connection for a specific user. If You use more than one sessions for the same username, You cannot disconnect the user properly, because ChilliSpot doesn't support IP address based remote disconnection method (only usernames are supported). So, always use simultaneous-use = 1 for ChilliSpot users (it can be defined in ACP / Edit users form).

Radius Manager supports the latest ChilliSpot 1.1.0. It is freely available on various websites and it is also downloadable from www.dmasoftlab.com.

You can build it from sources easily. To successfully install and configure ChilliSpot on your Linux host, You need the following hardware and software components:

- Linux host
- Two Ethernet interfaces (one for backbone and one for Hotspot clients)
- C/C++ development system

ChilliSpot installation steps

1. Download the ChilliSpot source archive on your host and decompress it:

```
[root@localhost]# gzip –d chillispot-1.1.0.tar.gz
[root@localhost]# tar xvf chillispot-1.1.0.tar
```

2. Enter ChilliSpot folder and create the Makefile:

```
[root@localhost]# cd chillispot-1.1.0
[root@localhost]# ./configure
```

3. Build it with **make** command and install with **make install**:

```
[root@localhost]# make
[root@localhost]# make install
```

4. Copy the file *doc/chilli.conf* to */etc*.

5. Now You can test the ChilliSpot executable issuing the command:

```
[root@localhost]# chilli
```

If You get errors like

"*chillispot[8792]: chilli.c: 917: radiussecret must be specified*"

it is completely normal. You have to edit */etc/chilli.conf* before begin to use it.

6. Uncomment debug flags in line 9:

```
fg
```

Uncommenting this line,You ensure to run ChilliSpot in foreground mode. It is good for debugging purposes. When the system is fully working, You will comment out this line again.

7. Define the DNS server IP address in line 59:

```
dns1 192.168.0.3
```

It must be a reachable DNS server, otherwise You will be unable to log on to ChilliSpot, instead it will wait a long time for the DNS response. Install and configure a DNS server on your Linux host and define the Linux IP as the DNS server address.

8. Define RADIUS server addresses in line 113 and 120:

```
radiusserver1 192.168.0.3
radiusserver2 192.168.0.3
```

It is the address where FreeRadius is running. Use only one server at same time. Define the same IP in both lines.

You can install FreeRadius, Radius Manager and ChilliSpot on a same host, but multiple host installation is also realizable.

9. Uncomment and define the RADIUS secret in line 139:

```
radiussecret testing123
```

The secret must match the one which is defined in ACP NAS definition. Don't forget, You have to restart FreeRadius server every time after modifying the NAS definitions in *raddb/clients.conf*. Unfortunately, FreeRadius doesn't read the NAS database at run-time.

10. Define RADIUS NAS IP in line 149. It is important to send the NAS IP in every RADIUS request for NAS identification.

```
radiusnasip 192.168.0.3
```

11. Define UAM server in line 237:

```
uamserver https://192.168.182.1/cgi-bin/hotspotlogin.cgi
```

The default gateway address is 192.168.182.2 for ChilliSpot, so don't change it. A working, HTTPS capable web server is required to serve the CGI versions of Chillispot login page.

12. Uncomment line 248 and define the UAM secret:

```
uamsecret secret
```

This secret must match the defined one in *hotspotlogin.cgi*.

13. Copy the *hotspotlogin.cgi* to HTTP server's *cgi-bin* folder. On Fedora it is */var/www/cgi-bin*. The file *hotspotlogin.cgi* must be executable, so modify the permission using *chmod*:

```
[root@localhost]# chmod 755 /var/www/cgi-bin/hotspotlogin.cgi
```

Completing this step You have finished configuring ChilliSpot. Now You have to set up a dedicated Ethernet interface in your Linux host for Hotspot users. As it was defined before, You need at least two network interface cards (NIC) installed in your host:

1. **Wan** – for connecting to the Internet.
2. **Lan** – for connecting the ChilliSpot Hotspot clients.

The Hotspot interface (lan) requires a special setup:

- Turn off all DHCP servers listening on that interface
- Do not assign any IP address to it

The correct ifcfg-xxx file looks like this:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
#IPADDR=192.168.182.1
#NETMASK=255.255.255.0
HWADDR=00:30:4F:03:DF:93
```

In this example we have commented out the IP address and netmask definitions of interface eth1. Create a similar ifcfg-xxx file on your system. After that restart the network on the Linux host.

When You Issue the command *ifconfig*, You have to see similar output to this:

```
eth1      Link encap:Ethernet  HWaddr 00:30:4F:03:DF:93
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:10 Base address:0x2000
```

If the output is correct, You can start using ChilliSpot. Start it with the following parameters:

```
[root@localhost]# chilli --coaport 3779
```

The required parameters are:

--coaport – Defines the port for the incoming disconnect requests (POD). Use value 3779 for your ChilliSpot server.

After ChilliSpot has been started, the connected machines have to get IP address from ChilliSpot server. You have to see the IP requests on the debug screen.

When You enter any address in the browser and the DNS server is working properly, You have to see the ChilliSpot login page within 2-3 seconds.

To ensure IP packets are forwarded properly to ChilliSpot interface, You have to enable the IP packet forwarding in Linux. You can do this with the following command:

```
[root@localhost]# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Also, masquerade the local Hotspot addresses:

```
[root@localhost]# iptables -t nat -A POSTROUTING -s 192.168.182.0/255.255.255.0 -j
MASQUERADE
```

Be sure You enter the line above without line breaks. In this example the Hotspot address range is **192.168.182.0/24**.

Now configure Radius Manager, define NASes (*raddb/clients.conf,* ACP) and begin using your newly installed ChilliSpot Hotspot system.

Cisco

Radius Manager supports the following features on Cisco NAS:

1. Authentication and authorization of PPP users (PPPoE, PPPtP, L2tP).
2. Bandwidth limitation per user (upload and download).
3. Automatic disconnection of expired accounts.
4. Limit simultaneous connections.
5. Static IP addresses.

Prerequisites are to have the correct IOS version in your Cisco NAS which can handle AAA new model and PPPoE, PPtP connections (vpdn-group or bba-group).

In this chapter we describe the RADIUS specific Cisco configuration entries. To enable AAA feature on Cisco, define the following entries using the configuration mode:

```
aaa new-model
aaa authentication ppp default group radius local
aaa authorization network default group radius
aaa accounting delay-start
aaa accounting update periodic 1
aaa accounting network default start-stop group radius
aaa pod server auth-type any server-key testing123

virtual-profile aaa
vpdn enable
vpdn-group pppoe
 accept-dialin
  protocol pppoe
  virtual-template 1

interface FastEthernet0/0
 ip address 192.168.0.98 255.255.255.0
 ip nat outside
 duplex auto
 speed auto

interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
 pppoe enable

interface Virtual-Template1
 ip unnumbered FastEthernet0/0
 ip nat inside
 peer default ip address pool pool1
 ppp authentication pap chap ms-chap
 ppp ipcp dns 192.168.0.3

ip local pool pool1 10.5.7.1 10.5.7.254
ip nat inside source list 1 interface Virtual-Template1 overload
access-list 1 permit 10.5.7.0 0.0.0.255

radius-server host 192.168.0.3 auth-port 1812 acct-port 1813
radius-server key testing123
```

The described configuration controls the AAA features on Cisco NAS. You have to set up the proper IP pools for local or public addresses, define NATing of local addresses etc. In the example above we are using DNS server address 192.168.0.3 and RADIUS server address 192.168.0.3. Substitute them with your own hosts. Also define the proper Ethernet interface names.

If You are using PPPoE connections, set up the correct interface to listen to PPPoE calls (pppoe enable).

This sample setup enables PPPoE server on FastEthernet0/1, enables POD packets and defines 1 minute interim update interval. The IP addresses assigned to PPPoE clients are defined in *pool1*. NATing is also enabled for the local IP addresses.

On Cisco, Radius Manager supports two types of bandwidth limitation:

1. **rate-limit**
2. **policy-map**

You can use the following commands on Cisco to check the actual bandwidth limitations of connected users:

show interfaces rate-limit
show policy-map interface
show policy-map session

Example of **show interfaces rate-limit** command:

```
Cisco2611#show interfaces rate-limit
Virtual-Access4
  Input
    matches: all traffic
      params:  128000 bps, 24576 limit, 49152 extended limit
      conformed 2 packets, 432 bytes; action: transmit
      exceeded 0 packets, 0 bytes; action: drop
      last packet: 369ms ago, current burst: 0 bytes
      last cleared 00:00:00 ago, conformed 6000 bps, exceeded 0 bps
  Output
    matches: all traffic
      params:  520000 bps, 98304 limit, 196608 extended limit
      conformed 0 packets, 0 bytes; action: transmit
      exceeded 0 packets, 0 bytes; action: drop
      last packet: 217264ms ago, current burst: 0 bytes
      last cleared 00:00:00 ago, conformed 0 bps, exceeded 0 bps
```

Some IOS versions don't support rate-limit method. If the bandwidth limitation isn't working with rate-limit method, define the policy-map on Cisco (both for upload and download) and define the same policy-map names in ACP / Edit service.

An example Cisco policy-map looks like this:

```
policy-map POLICY_UP_1024
  class class-default
   police cir 1128000 bc 192000 be 192000
     conform-action transmit
     exceed-action drop

policy-map POLICY_DOWN_1024
  class class-default
   police cir 1128000 bc 256000 be 256000
     conform-action transmit
     exceed-action drop
```

Example of **show policy-map interface** command:

```
Cisco2611#show policy-map interface
Virtual-Access3.2

  Service-policy input: 128

    Class-map: class-default (match-any)
      4 packets, 632 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      police:
          cir 128000 bps, bc 4000 bytes
        conformed 4 packets, 632 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

  Service-policy output: 512

    Class-map: class-default (match-any)
      1 packets, 16 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      police:
          cir 512000 bps, bc 16000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps
```

You can alternatively try **show policy-map session** command:

```
Cisco2611#show policy-map session
```

This is not a complete Cisco configuration guide. You can find more information on Cisco website: http://www.cisco.com.

StarOS

In current version of Radius Manager there is a limited support for StarOS v2 and v3 systems. The supported services are:

- PPPoE full support
- Limited acccess list support

Using PPPoE system You can easily build small and medium sizes ISP's. PPPoE is a reliable, industry standard authentication method for broadband connections.

We recommend to use Star v2 server edition. With StarOS, You cannot use more than one simultaneous connections for a specific user, otherwise You cannot disconnect the users properly, because StarOS PPPoE system doesn't support remote disconnect method based on IP address. In StarUtil the only one supported reference is the username. So, always use simultaneous-use = 1 for StarOS clients (it can be defined in ACP / Edit users form).

To use Radius Manager with PPPoE system in StarOS, You have to:

1. Set the specific interface to listen to PPPoE request
2. Enable and configure PPPoE server
3. Acivate PPPoE server at run-time
4. Set up RADIUS authentication
5. Configure firewall
6. Save and activate settings

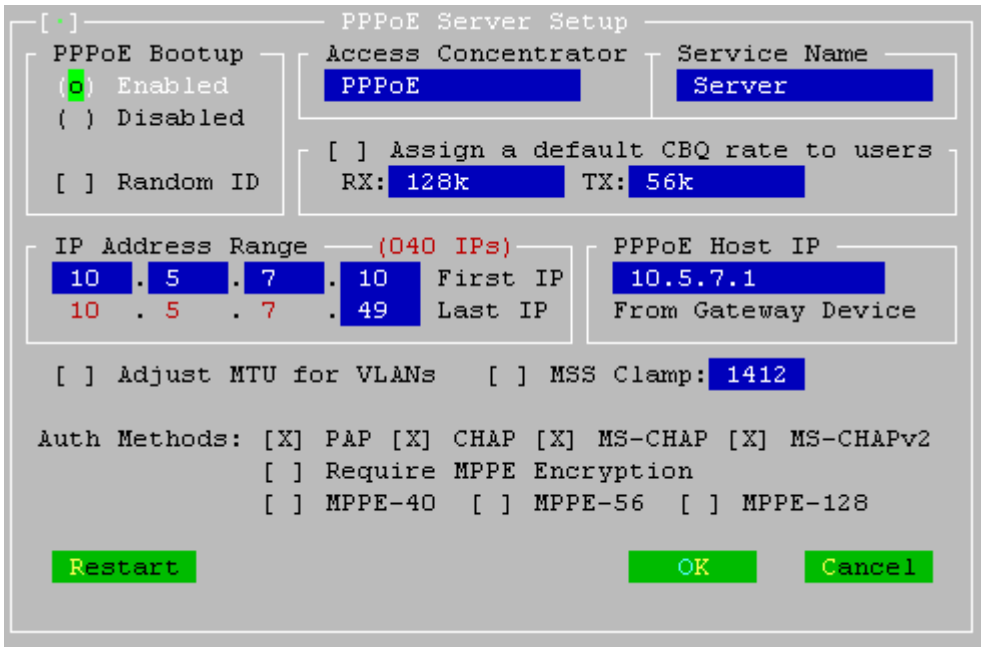
PPPoE server setup

1. Use option **interfaces / [interface name] / listen to pppoe requests: yes** to configure the interface to act as a PPPoE server interface.

2. PPPoE server configuration dialog can be invoked using the option:

   services / pppoe server / bootup/configuration settings

   The configuration of PPPoE server can be similar to this:



In this example we use PPPoE client pool 10.5.7.10 – 10.5.7.49. These addresses will be assigned to PPPoE clients. The PPPoE server IP is 10.5.7.1.

Select the compatible authentication method with your CPEs. PAP is unencrypted, so the recommended authentication methods are: **CHAP**, **MS-CHAP** and **MS-CHAP v2**. For compatibility You can also enable **PAP**.

3. You can control the PPPoE service activity without rebooting the system using the dialog:

   services / pppoe server / service activation

4. Set up RADIUS authentication using the option:

   services / pppoe server / radius authentication setup

   In this dialog define the following parameters (assuming your RADIUS server's IP address is 192.168.0.3 and using standard RADIUS ports):

   - authserver 192.168.0.3:1812
   - acctserver 192.168.0.3:1813
   - secret 192.168.0.3 testing123

   These three parameters are a must have. You can also edit retries, timeout etc.

5. If You are using local addresses for PPPoE clients, You have to masquerade them. Invoke the NAT editor using the option:

   advanced / scripts (cbq, firewall, nat, static arp, ...) / nat and static nat (1:1 ip mapping)

6. You can do this adding a new line to NAT / Static NAT table:

```
masq from 10.5.7.0/24 to dev ether1
```

   In this example the whole class C **10.5.7.0/24** is masqueraded to the backbone interface **ether1**. Always use the correct backbone interface.

   Save the settings and activate the changes.

7. Use option **file / activate changes** to save all your settings and activate PPPoE server on StarOS. Also activate the script changes using option

   advanced / scripts (cbq, firewall, nat, static arp, ...) / activate script changes

You have now successfully set up the PPPoE server on StarOS v2. Add the StarOS NAS in Radius Manager ACP, restart FreeRadius in debug mode and begin testing the PPPoE functionality.

Wireless access list setup

Radius Manager has limited compatibility with StarOS access list entries.

Unfortunately, when a wireless client gets connected using RADIUS access list, StarOS doesn't send only access request, but it also sends the accounting information for the access list user. It will not update the accounting information in regular intervals like PPPoE server, so You will see the access list user entry in ACP online users list, but with incorrect accounting data. So pay attention when using this feature.

To enable access list support, use access list editor for the specific interface. Invoke it using option:

wireless / [interface name] / access control list editor

Define the default action for handling wireless clients.

```
default = radius
```

Activate the changes. When a client tries to connect to StarOS WLAN interface, StarOS sends the access-request message to RADIUS server. It must respond with access-accept to allow the client to communicate with StarOS server.

Notes on StarOS compatibility

- Radius Manager is **fully compatible** with StarOS PPPoE server.
- Radius Manager has **limited compatibility** with StarOS access list system.
- Radius Manager is **not compatible** with StarOS Hotspot system. StarOS uses a stripped down version of ChilliSpot and it sends improper NAS IP address, doesn't accept the remote disconnect messages (POD), it sends accounting data in wrong format (upload and download are exchanged) and doesn't update the accounting data in regular intervals.

If You need a working and free Hotspot system, use ChilliSpot 1.1.0 on Linux. It supports all the features which are missing from StarOS and Radius Manager has full support for it.

pfSense

Radius Manager v 3.8 and newer versions include support for pfSense NAS. pfSense has a built in Chillispot captive portal which is fully controllable with Radius Manager.

The following features are available:

- Authentication
- Accounting
- Bandwidth shaping per individual users
- Download traffic limitation
- Upload traffic limitation
- Combined traffic traffic limitation
- Online time limitation
- Account expiry

Restrictions:

- pfSense **dos not support remote disconnection** using POD packets, instead it is using reauthentication which has drawbacks against the POD system.
- Because pfSense uses reauthentication method to check the validity of the logged on account, at least **sim-use = 2** has to be set for every pfSense user in Radius Manager ACP / Edit user dialog. Sim-use = 1 will result immediately disconnection of the user when the first reauthentication packet is sent to the RADIUS server (RADIUS server thinks the user is already online and doesn't give a permission for a new concurrent connection which causes pfSense to close the active session of the current user).

This installation manual is not a complete pfSense user's manual. It covers the most important and RADIUS specific configuration steps only. For more pfSense informations visit their official web site: http://www.pfsense.com

To configure pfSense as captive portal You have to complete the following steps:

1. Configure **interfaces** (WAN and LAN)
2. Configure **DNS**
3. Configure **DHCP server**
4. Configure **captive portal**

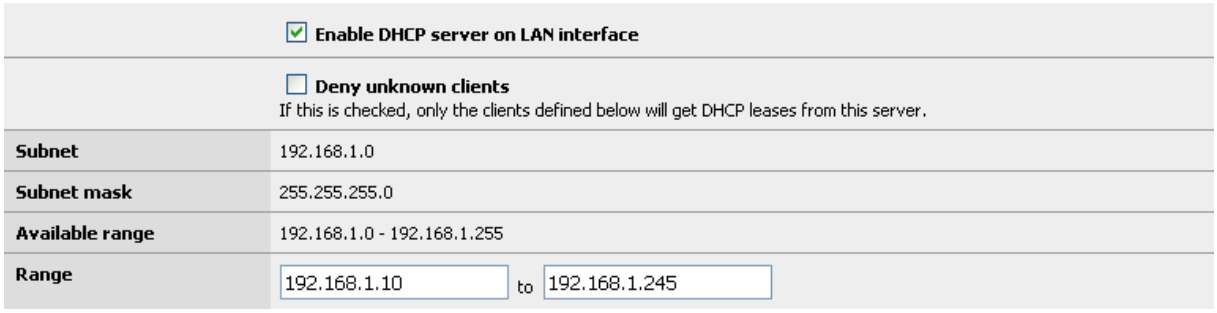
Configuring the network interfaces and DNS

Use the configuration console set the following parameters of the pfSense router:

1. **WAN address –** Use static address, Radius Manager can communicate with the NAS if it is using static IP address.
2. **LAN address –** It is the gateway of your Hotspot clients. In our example it is 192.168.1.1 with subnet /24.
3. **Default gateway** – Set the correct gateway to reach the world.
4. A valid **DNS server** address – Set it using the web configurator or the configuration console.

Configuring the DHCP server

Open the dialog in WEB configurator using the menu **Services / DHCP server**. Enter the valid network range and enable the DHCP server on the LAN interface as it is shown on the picture below. Be sure the LAN IP address is located in the same subnet.

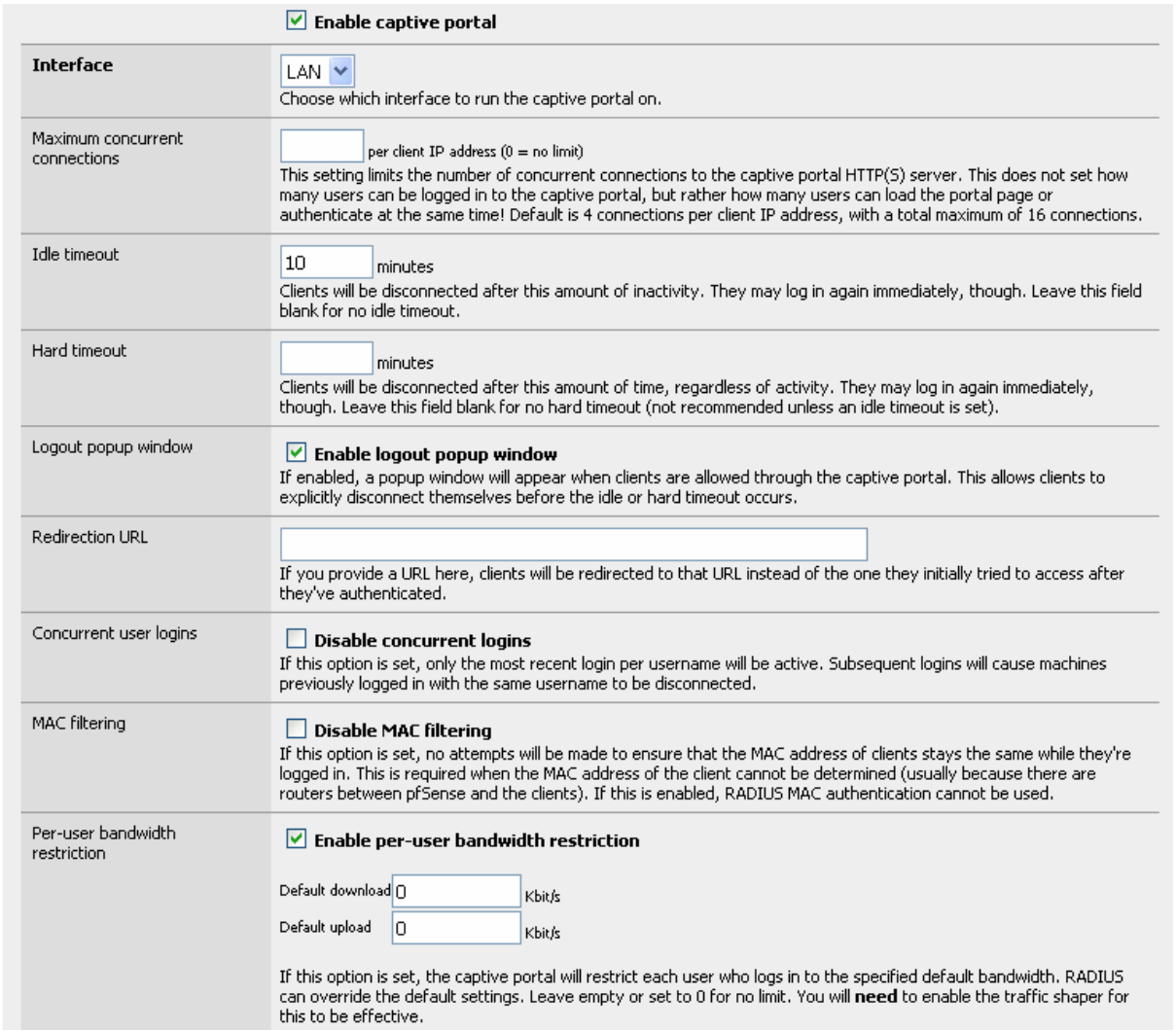


	☑ **Enable DHCP server on LAN interface**
	☐ **Deny unknown clients** If this is checked, only the clients defined below will get DHCP leases from this server.
Subnet	192.168.1.0
Subnet mask	255.255.255.0
Available range	192.168.1.0 - 192.168.1.255
Range	192.168.1.10 to 192.168.1.245

Configuring the captive portal

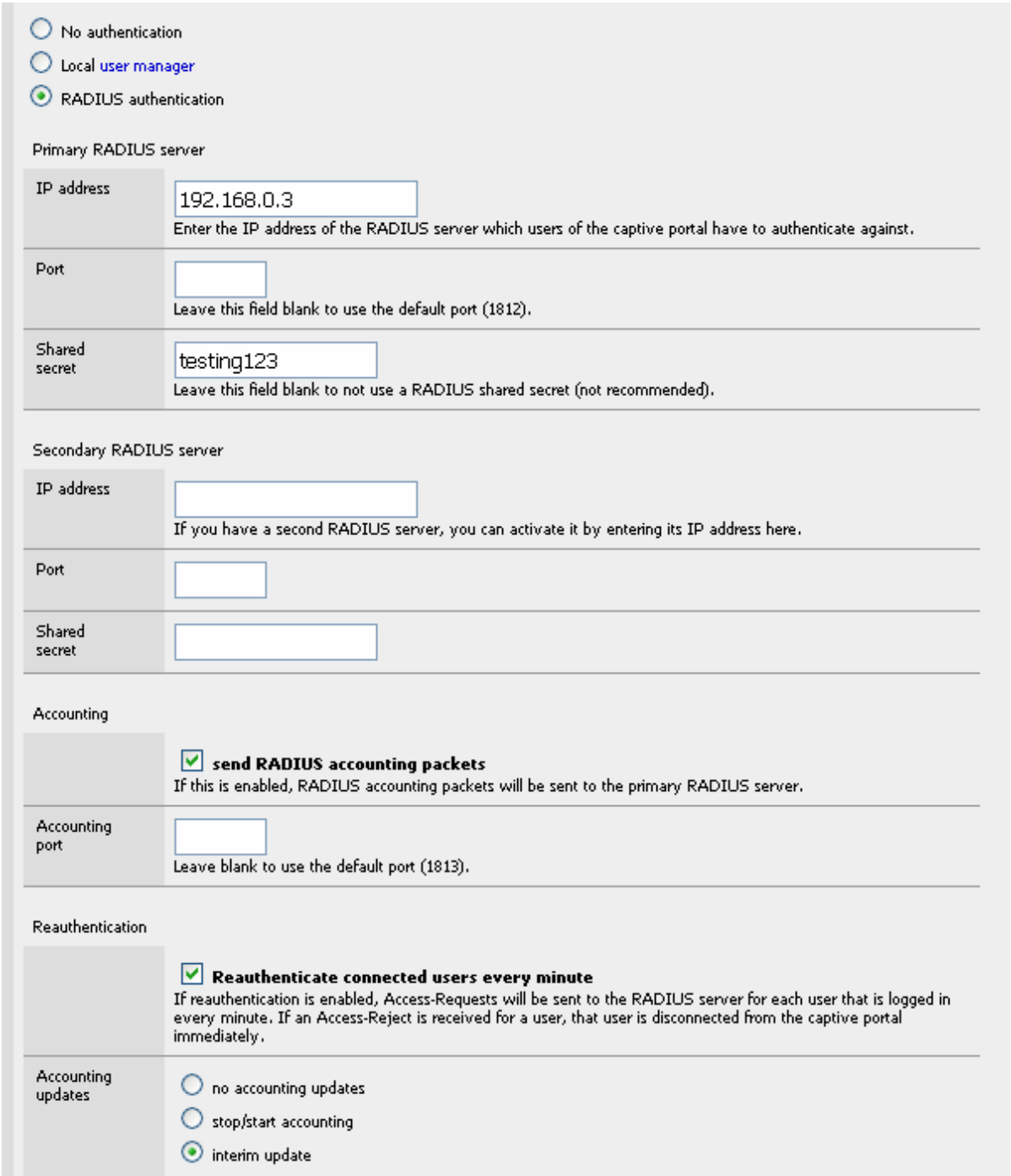
Follow these simple steps to enable and configure the captive portal with RADIUS support:

1. Open the Captive portal options (Services / Captive portal)
2. Enable the captive portal with checkbox
3. Select the interface where You will connect the Hotspot clients
4. Set idle timeout to 10 minutes
5. Enable logout popup window with checkbox
6. Enable per-user bandwidth restriction
7. Select RADIUS authentication



	☑ **Enable captive portal**
Interface	LAN Choose which interface to run the captive portal on.
Maximum concurrent connections	per client IP address (0 = no limit) This setting limits the number of concurrent connections to the captive portal HTTP(S) server. This does not set how many users can be logged in to the captive portal, but rather how many users can load the portal page or authenticate at the same time! Default is 4 connections per client IP address, with a total maximum of 16 connections.
Idle timeout	10 minutes Clients will be disconnected after this amount of inactivity. They may log in again immediately, though. Leave this field blank for no idle timeout.
Hard timeout	minutes Clients will be disconnected after this amount of time, regardless of activity. They may log in again immediately, though. Leave this field blank for no hard timeout (not recommended unless an idle timeout is set).
Logout popup window	☑ **Enable logout popup window** If enabled, a popup window will appear when clients are allowed through the captive portal. This allows clients to explicitly disconnect themselves before the idle or hard timeout occurs.
Redirection URL	If you provide a URL here, clients will be redirected to that URL instead of the one they initially tried to access after they've authenticated.
Concurrent user logins	☐ **Disable concurrent logins** If this option is set, only the most recent login per username will be active. Subsequent logins will cause machines previously logged in with the same username to be disconnected.
MAC filtering	☐ **Disable MAC filtering** If this option is set, no attempts will be made to ensure that the MAC address of clients stays the same while they're logged in. This is required when the MAC address of the client cannot be determined (usually because there are routers between pfSense and the clients). If this is enabled, RADIUS MAC authentication cannot be used.
Per-user bandwidth restriction	☑ **Enable per-user bandwidth restriction** Default download 0 Kbit/s Default upload 0 Kbit/s If this option is set, the captive portal will restrict each user who logs in to the specified default bandwidth. RADIUS can override the default settings. Leave empty or set to 0 for no limit. You will **need** to enable the traffic shaper for this to be effective.

8. Enter the primary RADIUS server IP address
9. Enter the shared secret
10. Turn on checkbox “send RADIUS accounting packets”
11. Turn on checkbox “Reauthenticate connected users every minute”
12. Select accounting updates “interim update”



CTS setup

Radius Manager has a special feature: the **Connection Tracking System**. It is available only in Radius Manager CTS version or higher. With the help of it the system can track and log all the TCP and UDP connections for all registered (online) users.

By default when You install the CTS enabled version of Radius Manager, it will use the default CTS database (CONNTRACK). It is strongly recommended to use a **separate database** host for the CONNTRACK database, due to the enormous amount of data stored daily. It can be even a 100-500 MegaBytes / day. Fast disks are also recommended to be able to seek and store the data in real time. Radius Manager periodically stores the traffic data to CONNTRACK database (typically in every 5–60 seconds).

To use the CTS feature You need a Mikrotik router. It can be:

1. the same router where the PPP and Hotspot users are connected to or
2. a separate router which passes traffic through on it (backbone router)

If You use the second option, You can't masquerade the clients on PPP / Hotspot server and cannot use transparent proxy on it. You must ensure the packets are going through the traffic logger Mikrotik with their original IP addresses. Masquerading can be done after the packets were processed by the CTS logger router.

When the packets are going through the logger router, the router processes them using a firewall rule and sends the log data to the Radius Manager CTS host.

The following configuration has to be set up on the logger Mikrotik router:

1. Add the following rule to the filter table:

```
/ip firewall filter add chain=forward src-address=10.5.7.0/24 protocol=tcp \
connection-state=new action=log

/ip firewall filter add chain=forward src-address=10.5.7.0/24 protocol=udp \
connection-state=new action=log
```

It will log all UDP and TCP packets going through the logger router.

2. Enable remote logging for firewall events:

```
/system logging action add name=remote1 remote=192.168.0.3:4950 target=remote

/system logging add topics=firewall action=remote1
```

Test the logging system by executing the **rmconntrack** binary on Linux in debug mode:

```
[root@localhost]# rmconntrack -x
rmconntrack daemon started successfully.
```

When online user's UDP or TCP traffic is going through the logger Mikrotik, You have to see the logging data arriving to Linux.

ADDITIONAL SETUP

Log files

FreeRadius log file sometimes became enormously big (10-30 MBs), and the Linux file system is unable to handle it fast enough which is required for a flawless work of FreeRadius server. It can cause degraded system performance and / or RADIUS timeouts. To prevent such problems, the log file has to be stripped regularly.

To set up automatic log rotation for **radiusd.log**, simply copy the file *etc/logrotate/radiusd* from radiusmanager tar archive to */etc/logrotate.d* folder on your Linux host. The automatic installer also does the same job. The included logrotate script is Redhat and Debian compatible. It can also be used on other systems with minor modifications.

Starting daemons at boot time

Radius Manager system supports auto startup for daemons **radiusd**, **rmpoller** and **rmconntrack**.

The appropriate init scripts have to be installed in */etc/init.d* folder. The automatic installer installs the **rmpoller** and **rmconntrack** init scripts, but **radiusd** script has to be copied manually. If You are using manual method, copy the **rmpoller**, **rmconntrack** and **[debian]/radiusd** or **[redhat]/radiusd** from Radius Manager installation archive to */etc/init.d* folder.

After copying the appropriate *radiusd* script to */etc/init.d*, change its permissions to 755:

```
[root@localhost]# chmod 755 /etc/init.d/radiusd
```

The following methods are available to set up automatic service startup:

- use Webmin to start services at boot time or
- create symbolic links to appropriate runlevels or
- use command **chkconfig --add [service_name]** on Fedora

Chkconfig example:

```
[root@localhost]# chkconfig --add radiusd
[root@localhost]# chkconfig --add rmpoller
[root@localhost]# chkconfig --add rmconntrack
```

Rootexec permission problems

On some Linux systems due to the system security Radius Manager installer is unable to set the 4755 permission on **rootexec** binary. In that case the manual method has to be used:

```
[root@localhost]# chmod 4755 /usr/local/sbin/rootexec
```

Remote UNIX host synchronization

To use the remote UNIX host user synchronization with RADIUS users, passwordless SSH login is required to be set on the remote host.

OpenSSH server – the host which is **synchronized** (the email server)
OpenSSH client – the Radius Manager server which **synchronizes** the remote host

The following steps have to be followed in order to set up the passwordless SSH login successfully.

1. Generate your OpenSSH protocol 2 RSA key:

```
[root@localhost]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
8c:5f:0c:ea:8a:e6:dd:a0:45:d6:e9:42:3e:9a:5a:95 root@dtk.localdomain
```

Answer with enter to every question. Use empty passphrase and use the default file name for key.

2. Append the contents of your public key to the *authorized_keys* file on the remote OpenSSH server:

```
[root@localhost]# cat ~/.ssh/id_rsa.pub | ssh 192.168.0.4 "cat - >>
~/.ssh/authorized_keys"
root@192.168.0.4's password:
```

Where 192.168.0.4 is the remote server. When it is asking for the root password of the remote server You have to enter the proper password. The *.ssh* subfolder must exists on the remote host in */root* folder before issuing the previous command. Create the *.ssh* folder manually if required.

After finishing this operation, You can test the passwordless SSH access to the remote server using the following remote ls command:

```
[root@localhost]# ssh 192.168.0.4 ls
download
install
mail
work
```

REFERENCE

Radius Manager configuration files

system_cfg.php

The file *system_cfg.php* is located in *radiusmanager/config* folder. The main configuration entries are:

```
// database credentials

define("db_host", "localhost");          // database host
define("db_base", "radius");             // database name
define("db_user", "radius");             // database user
define("db_psw", "radius123");           // database password
define("db_host_ct", "localhost");       // database host
define("db_base_ct", "conntrack");       // database name
define("db_user_ct", "conntrack");       // database user
define("db_psw_ct", "conn123");          // database password
```

- **db_host** – RADIUS MySql database hostname or IP address.
- **db_base** – RADIUS MySql database name.
- **db_user** – RADIUS MySql database username.
- **db_psw** – RADIUS MySql database password.
- **db_host_ct** – CONNTRACK MySql database hostname or IP address.
- **db_base_ct** – CONNTRACK MySql database name.
- **db_user_ct** – CONNTRACK MySql database username.
- **db_psw_ct** – CONNTRACK MySql database password.

```
// system paths and files

define("radman_dir", "/var/www/html/radiusmanager");
define("raddb", "/usr/local/etc/raddb");
define("clients_conf", "clients.conf");
define("lang_dir", "lang");
```

- **radman_dir** – Define the absolute path of Radius Manager HTML files.
- **raddb** – The full path of raddb directory.
- **clients_conf** – The name of clients.conf file.
- **lang_dir –** Folder name of language files.

```
// system definitions

define("admin_user", "admin");
define("rootexec_psw", "12345");
define("rmscheduler_psw", "12345");
define("nas_port_mt", 1700);
define("nas_port_chilli", 3779);
define("nas_port_cisco", 1700);
define("hotspot_ip", "http://10.5.7.1");
define("no_limit_date", "2020-12-31");
define("max_card_quantity", 10000);
define("cardnum_integers", 6);
define("card_pin_len", 8);
define("card_psw_len", 4);
define("ias_pin_length", 8);
define("ias_psw_length", 4);
```

```
define("rndchars", "0123456789");
define("rndstring_len", 4);
define("max_smsnums", 3);
define("max_pinfails", 3);
define("max_verifyfails", 3);
define("quickjump_max_pages", 10);
define("rows_per_page", 100);
define("csv_max_rows", 1000000);
define("cc_years", 5);
define("smtp_relay", "localhost");
define("mail_from", "admin@myisp.com");
define("mail_reply", "admin@myisp.com");
define("mail_preview", "admin@myisp.com");
define("mail_localdomain", "localhost.localdomain");
define("regexp_username", '/^[a-z0-9._]+$/');
define("regexp_managername", '/^[a-z0-9._]+$/');
define("regexp_mac", '/^[:a-z0-9._]+$/');
define("regexp_psw", '/^[a-zA-Z0-9._]+$/');
```

- **admin_user** – The name of the Radius Manager superuser.
- **rootexec_psw** – Defines the password for rootexec program. It has to be equivalent with that which is defined in */etc/radiusmanager.cfg.*
- **rmscheduler_psw** – This password match the one entered in crontab.
- **nas_port_mt –** Radius incoming packet port for Mikrotik. It is global for all Mikrotik NASes.
- **nas_port_chilli –** Radius incoming packet port for ChilliSpot. It is global for all ChilliSpot NASes.
- **nas_port_cisco –** Radius incoming packet port for Cisco. It is global for all Cisco NASes.
- **hotspot_ip** – The address of the Hotspot server for http redirections.
- **no_limit_date** – Use this date for unlimited Unix account expiration.
- **max_card_quantity –** The maximum number of cards which can be generated at once.
- **cardnum_integers –** How many serial numbers digits to show when You list card codes (first column).
- **card_pin_len –** PIN length of prepaid cards.
- **card_psw_len –** Password length of prepaid cards.
- **ias_pin_length –** IAS username length.
- **ias_psw_length –** IAS password length.
- **rndchars –** Characters in account verification code.
- **rndstring_len –** Length of verification code.
- **max_smsnums** – Maximal number of card verification SMS.
- **max_pinfails** – Maximal number of wrong PIN codes.
- **max_verifyfails** – Maximal number of verification failures.
- **quickjump_max_pages** – How many pages to display in quickjumps.
- **rows_per_page** – Table rows per page.
- **csv_max_rows** – Number of rows in CSV file.
- **cc_years** – How many years to display in CC expiration listboxes.
- **smtp_relay** – SMTP relay.
- **mail_from** – Warning email sender.
- **mail_reply** – Warning email return path.
- **mail_preview** – Preview user of mass mail.
- **mail_localdomain** – Warning email local domain.
- **regexp_username** – Regular expression for validating user names.
- **regexp_managername** – Regular expression for validating manager names.
- **regexp_mac** – Regular expression for validating MACs.
- **regexp_psw** – Regular expression for validating passwords.

```
// CTS specific data

define("keep_connlog", 7);
define("keep_syslog", 30);
define("keep_actsrv", 60);
```

- **keep_connlog** – How many days to keep the connection log data.
- keep_syslog – How many days to keep the system log data.
- keep_actsrv – How many minutes to keep the actual service data.

```
// limits

define("min_username_len", 4);
define("max_username_len", 32);
define("mac_username_len_mikrotik", 17);
define("mac_username_len_staros", 12);
define("min_psw_len", 4);
define("max_psw_len", 32);
define("hsmac_min_psw_len", 0);
define("hsmac_max_psw_len", 32);
define("mobile_minlen", 10);
define("mobile_maxlen", 11);
```

- **min_username_len –** Define the minimal allowed length of the user name for the new user.
- **max_username_len –** Define the maximal allowed length of the user name for the new user.
- **mac_username_len_mikrotik –** Define the length of the Mikrotik MAC user name.
- **mac_username_len_staros –** Define the length of the StarOS MAC user name.
- **min_psw_len –** Define the minimal allowed password length.
- **max_psw_len –** Define the maximal allowed password length.
- **hsmac_min_psw_len –** Minimal password length of Hotspot MAC users.
- **hsmac_max_psw_len –** Maximal password length of Hotspot MAC users.
- **mobile_minlen –** Minimal allowed length of mobile number (verification).
- **mobile_maxlen –** Maximal allowed length of mobile number (verification).

```
// card PDF export

define("username_x_pos", 45);
define("username_y_pos", 36);
define("psw_x_pos", 45);
define("psw_y_pos", 44);
define("pin_x_pos", 33);
define("pin_y_pos", 40);
define("price_x_pos", 67);
define("price_y_pos", 19);
define("date_x_pos", 53);
define("date_y_pos", 53);
define("user_font_type", "Arial");
define("user_font_size", 14);
define("user_font_color", "000000");
define("date_font_type", "Arial");
define("date_font_size", 10);
define("date_font_color", "000000");
define("price_font_type", "Arial");
define("price_font_size", 14);
define("price_font_color", "FFF7A1");
define("card_left_margin", 13);
define("card_top_margin", 13);
define("card_classic_bg_filename", "classic_bg.png");
define("card_refill_bg_filename", "refill_bg.png");
define("card_bg_width", 85);
define("card_bg_height", 50);
```

- **username_x_pos** – User name x position on classic prepaid card.
- **username_y_pos** – User name y position on classic prepaid card.
- **psw_x_pos** – Password x position on classic prepaid card.

- **psw_y_pos** – Password y position on classic prepaid card.
- **pin_x_pos** – PIN x position on refill card.
- **pin_y_pos** – PIN y position on refill card.
- **price_x_pos** – Price x position on card.
- **price_y_pos** – Price y position on card.
- **date_x_pos** – Valid till x position on card.
- **date_y_pos** – Valid till y position on card.
- **user_font_type** – PIN / password font typeface.
- **user_font_size** – PIN / password font size.
- **user_font_color** – PIN / password font color.
- **date_font_type** – Date font typeface.
- **date_font_size** – Date font size.
- **date_font_color** – Date font color.
- **price_font_type** – Price font typeface.
- **price_font_size** – Price font size.
- **price_font_color** – Price font color.
- **card_left_margin** – Left margin.
- **card_top_margin** – Top margin.
- **card_classic_bg_filename** – Classic prepaid card background picture file.
- **card_refill_bg_filename** – Refill card background picture file.
- **card_bg_width** – Prepaid card background picture width.
- **card_bg_height** – Prepaid card background picture height.

```
// unix executables

define("cmd_rootexec", "/usr/local/sbin/rootexec");
define("cmd_radclient", "/usr/local/bin/radclient");
define("cmd_starutil", "/usr/local/bin/starutil");
define("cmd_useradd", "/usr/sbin/useradd");
define("cmd_userdel", "/usr/sbin/userdel");
define("cmd_chmod", "/usr/bin/chmod");
define("cmd_usermod", "/usr/sbin/usermod");
define("cmd_passwd", "/usr/sbin/passwd");
define("cmd_edquota", "/usr/sbin/edquota");
```

- **cmd_rootexec** – rootexec executable with full path.
- **cmd_radclient** – radclient executable with full path.
- **cmd_starutil** – starutil executable with full path.
- **cmd_useradd** – useradd executable with full path.
- **cmd_userdel** – userdel executable with full path.
- **cmd_chmod** – chmod executable with full path.
- **cmd_usermod** – usermod executable with full path.
- **cmd_passwd** – passwd executable with full path.
- **cmd_edquota** – edquota executable with full path.

paypal_cfg.php

Radius Manager supports **PayPal Express Checkout**, **PayPal Website Payments Pro** and **PayPal Website Payments Standard** API. PayPal Express Checkout works with premier and business accounts, but PayPal Website Payments Pro (CC processing) will work only with a Pro account or better and requires the merchant to be registered from US / UK. PayPal Website Payments Standard can be used for balance and CC payments and it supports multiple merchant countries.

PayPal subsystem configures via file *paypal_cfg.php* which is located in *radiusmanager/config* folder. The main configuration entries in *paypal_cfg.php* are:

```
// API credentials of PayPal Express Checkout and PayPal Website Payments Pro

define('API_USERNAME', 'username');
define('API_PASSWORD', 'password');
define('API_SIGNATURE', 'signatue');

// API credentials of PayPal Website Payments Standard

define("DEFAULT_USER_NAME", "username");
define("DEFAULT_PASSWORD", "password");

define("DEFAULT_EMAIL_ADDRESS", "info@mycompany.com");
define("DEFAULT_IDENTITY_TOKEN", "token");

define("DEFAULT_EWP_CERT_PATH", "certs/ewp-cert.pem");
define("DEFAULT_EWP_PRIVATE_KEY_PATH", "certs/ewp-key.pem");
define("DEFAULT_EWP_CERT_ID", "cert_id");
define("PAYPAL_CERT_PATH", "certs/paypal-cert.pem");

// enable sandbox test mode

define("TEST_MODE", TRUE);

// other

define("CC_MERCHANT_COUNTRY", "US");
```

Description of the configuration entries:

- **API_USERNAME** – API user name (Express Checkout and Website Payments Pro).
- **API_PASSWORD** – API password (Express Checkout and Website Payments Pro).
- **API_SIGNATURE** – API signature (Express Checkout and Website Payments Pro).
- **DEFAULT_USER_NAME** – API user name (Website Payments Standard).
- **DEFAULT_PASSWORD** – API password (Website Payments Standard).
- **DEFAULT_EMAIL_ADDRESS** – merchant email address to be displayed on PayPal site. (Website Payments Standard).
- **DEFAULT_IDENTITY_TOKEN** – API identity token (Website Payments Standard).
- **DEFAULT_EWP_CERT_PATH** – API certificate public key (Website Payments Standard).
- **DEFAULT_EWP_PRIVATE_KEY_PATH** – API certificate private key (Website Payments Standard).
- **DEFAULT_EWP_CERT_ID** – API certificate ID (Website Payments Standard).
- **PAYPAL_CERT_PATH** – PayPal certificate public key (Website Payments Standard).
- **TEST_MODE** – Set it to TRUE to use the Sandbox testing environment or false to use the real PayPal account.
- **CC_MERCHANT_COUNTRY** – US or UK, used for Website Payments Pro API.

For testing purposes use the PayPal Sandbox site. Create a testing account and define the credentials in *paypal_cfg.php*. When testing, be sure You are logged on to PayPal developer site all the time.

netcash_cfg.php

Radius Manager system supports NetCash (www.netcash.co.za) credit card payment gateway. You need a NetCash merchant account to use this feature.

NetCash module is configurable via *netcash_cfg.php* which is located in *radiusmanager/config* folder. The configuration entries in *netcash_cfg.php* are:

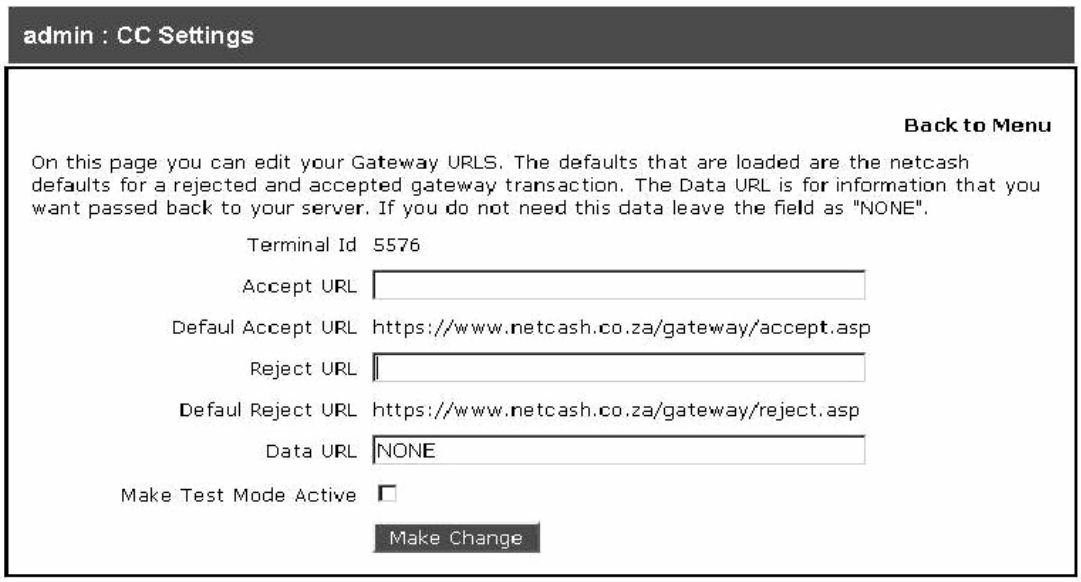
```
// Netcash credentials

define('NETCASH_USERNAME', 'username');
define('NETCASH_PASSWORD', 'password');
define('NETCASH_PIN', '12345');
define('TERMINAL_NUMBER', '12345');

// other data

define('NETCASH_EMAIL', 'info@mycompany.com');
```

Description of the configuration entries:

- **NETCASH_USERNAME** – NetCash merchant user name.
- **NETCASH_PASSWORD** – NetCash merchant password.
- **NETCASH_PIN** – NetCash PIN code.
- **TERMINAL_NUMBER** – NetCash terminal number.
- **NETCASH_EMAIL** – Email address to receive transaction reports sent by NetCash.

You have to define the Accept URL and Reject URL on Netcash.co.za site. Enter it in the following form:

http://yourhost/radiusmanager/netcash_return.php



authorizenet_cfg.php

From version 3.7 Radius Manager supports authorize.net to accept credit cards online. The system doesn't store any data on the host, insteam it simply forwards the CC data to authorize.net (AIM integration method). Be sure You are running the HTTP server in **secure mode** (SSL) when You are working with credit cards!

Authorize.net module is configurable via *authorizenet_cfg.php* which is located in *radiusmanager/config* directory. The configuration entries are:

```
// Authorize.net API Login ID and Transaction Key

define('AUTHORIZENET_USERNAME', 'login_id');
define('AUTHORIZENET_TRANSKEY', 'transaction_key');
define('AUTHORIZENET_URL',      'https://test.authorize.net/gateway/transact.dll');
```

Description of the configuration entries:

- **AUTHORIZENET_USERNAME** – API user name.
- **AUTHORIZENET_TRANSKEY** – API transaction key.
- **AUTHORIZENET_URL** – The gateway URL. For real situations enter the live authorize.net URL (https://secure.authorize.net/gateway/transact.dll).

dps_cfg.php

DPS Express Payment gateway (www.paymentexpress.com) is available in Radius Manager 3.8 to accept credit cards online. It supports mainly the New Zealand region. The system doesn't store any data on the host, the CC handling is done one the DPS site (redirection). When a CC has processed (success or failure) the browser gets directed back to Radius Manager site.

DPS module is configurable via *dps_cfg.php* which is located in *radiusmanager/config* directory. The main configuration entries are:

```
define("DPS_USERNAME",   "username");
define("DPS_KEY",        "key");

define("DPS_EMAIL",      "info@mycompany.com");
define("currency_dps",   "NZD");
```

Description of the configuration entries:

- **DPS_USERNAME** – API user name.
- **DPS_KEY** – API transaction key.
- **DPS_EMAIL** – The email address of the merchant.
- **currency_dps** – The accepted currency in the system (must be equal with the currency which is defined in ACP / system settings).

radiusmanager.cfg

The file *radiusmanager.cfg* is located in */etc* folder. It is the configuration file for the helper binaries. The content of *radiusmanager.cfg* is:

```
db_host             localhost                   ; mysql RADIUS host address
db_name             radius                      ; mysql RADIUS database name
db_user             radius                      ; mysql RADIUS username
db_psw              radius123                   ; mysql RADIUS password
db_host_cts         localhost                   ; mysql CONNTRACK host address
db_name_cts         conntrack                   ; mysql CONNTRACK database name
db_user_cts         conntrack                   ; mysql CONNTRACK username
db_psw_cts          conn123                     ; mysql CONNTRACK password
db_sock             /var/lib/mysql/mysql.sock   ; mysql main socket location
radman_path         /var/www/radiusmanager      ; Radius Manager full path
inactivity          10                          ; timeout in minutes to cleanup inactive sessions
rootexec_psw        12345                       ; rootexec password
poller_pause        60                          ; disconnect handler cycle pause in seconds
radclient           /usr/local/bin/radclient    ; radclient path
starutil            /usr/local/bin/starutil     ; starutil path
nas_port_mt         1700                        ; global POD port of Mikrotik
nas_port_chilli     3779                        ; global POD port of ChilliSpot
nas_port_cisco      1700                        ; global POD port of Cisco
smtp_relay          localhost                   ; smtp relay
mail_from           admin@myisp.com             ; email sender address
mail_reply          admin@myisp.com             ; email reply adddress
mail_localdomain    localhost.localdomain       ; email local domain
logger_port         4950                        ; port for accepting syslog messages from Mikrotik
connlog_pause       5                           ; connlog parser cycle pause
connlog_file        /tmp/rmconnlog              ; filename of temporary connection storage
pid_dir             /usr/local/var/run          ; directory of PID files
cts_threads         16                          ; number of thread for connection data processing
```

Description of the configuration entries:

- **db_host** – Define the RADIUS MySql database host.
- **db_name** – Define the RADIUS MySql database name.
- **db_user** – Define the RADIUS MySql database user.
- **db_psw** – Define the RADIUS MySql database password.
- **db_host_cts** – Define the CONNTRACK MySql database host.
- **db_name_cts** – Define the CONNTRACK MySql database name.
- **db_user_cts** – Define the CONNTRACK MySql database user.
- **db_psw_cts** – Define the CONNTRACK MySql database password.
- **db_sock** – Define the MySql socket location.
- **radman_path** – Define the Radius Manager full web path.
- **inactivity** – Define the timeout in minutes for automatically cleaning up the inactive accounting sessions.
- **rootexec_psw** – The password for executing *rootexec* binary.
- **poller_pause** – Define the time interval in seconds when *rmpoller* checks for the online users and calculates the limits. Use values **60 – 300** seconds. Using smaller values You will have more accurate disconnect precisity. Higher values enables the users to go into negative (Bytes, time).
- **radclient** – Full path of the *radclient* binary file.
- **starutil** – Full path of the *starutil* binary file.
- **nas_port_mt** – RADIUS POD port for all Mikrotik NASes in the system.
- **nas_port_chilli** – RADIUS POD port for all StarOS NASes in the system.
- **nas_port_cisco** – RADIUS POD port for all Cisco NASes in the system.
- **smtp_relay** – SMTP server IP address for the binaries. The IP address has to be resolvable in order to use it. Define it in */etc/hosts*
- **mail_from** – The email address to be displayed as sender.
- **mail_reply** – The email address replying emails.

- **mail_localdomain** – The domain name for creating email addresses for RADIUS users with unspecified email addresses. The final address will look like: radius_username@mail_localdomain
- **logger_port** – Define the listener port for syslog messages.
- **connlog_pause** – Define the interval for moving CTS data blocks.
- **connlog_file** – Define the temporary connection log file.
- **pid_dir** – Directory of PID files
- **cts_threads** – Number of thread for connection data processing.

Configuring PayPal Website Payments Standard API

From all PayPal APIs the PayPal Website Payments Standard API is the most complicated to configure. It has an advantage over the Pro API: it supports multiple merchant countries, not only US.

To successfully configure the API, You'll need:

1. Self signed SSL certificate
2. Configured preferences in your PayPal account
3. Properly entered credentials in paypal_cfg.php

Follow this description to set up your system to accept CC and PayPal payments online, using the PayPal Website Payments Standard API. For testing purposes we recommend to use the Sandbox. It is a great feature of PayPal, where You can experiment with various test accounts without charging a real Credit Card.

To enable the Sandbox mode set TRUE in the following line in paypal_cfg.php:

define("TEST_MODE", TRUE);

A premier or busines seller account is required to use the PayPal Website Payments Standard API.

Configuring the PayPal account

1. Request an API certificate.

- Log on to your PayPal account and click on **My account** / **Profile**
- Click the **API access**
- Click **Request API credentials**
- Click **Request API certificate**
- Copy the API username and password to paypal_cfg.php and click **Done**

```
define("DEFAULT_USER_NAME", "your API username");
define("DEFAULT_PASSWORD", "your API password");
```

2. Get your identity token.

- Go to **My account / Profile / Website payment preferences**.
- Turn on **Auto return**.
- Enter any **Return URL**. It can be the URL of your Radius Manager server. The URL name is not important, Radius Manager will overwrite this variable at run time in every PayPal request.
- Enable **Payment data transfer**.
- Click **Save**. When You return to the same page, You will see the identity token in **Payment data transfer**. Copy this token to paypal_cfg.php

```
define("DEFAULT_IDENTITY_TOKEN", "token");
```

3. Enter your PayPal account email address.

```
define("DEFAULT_EMAIL_ADDRESS", "premier@mycompany.com");
```

Generating SSL certificates

You need a SSL certificate to use the Website Payments Standard API. To generate a certificate follow these steps exactly:

1. Generating Your Private Key Using OpenSSL

Using the openssl program, enter the following command to generate your private key. The command generates a 1024-bit RSA private key that is stored in the file ewp-key.pem:

```
[root@localhost]# openssl genrsa -out ewp-key.pem 1024
```

2. Generating Your Public Certificate Using OpenSSL

The public certificate must be in PEM format. To generate your certificate, enter the following openssl command, which generates a public certificate in the file my-pubcert.pem:

```
[root@localhost]# openssl req -new -key ewp-key.pem -x509 -days 365 -out ewp-
cert.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:NY
Locality Name (eg, city) [Newbury]:Albany
Organization Name (eg, company) [My Company Ltd]:My Company
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:billing.myisp.com
Email Address []:info@myisp.com
```

Copy both files (ewp-cert.pem, ewp-key.pem) to *radiusmanager/certs* directory.

3. Uploading your public certificate to your PayPal account

- **Log in** to your **PayPal** Business or Premier account
- Click the **Profile** subtab.
- In the **Seller Preferences** column, click the **Encrypted Payment Settings** link. The Website Payment Certificates page appears.
- Scroll down the page to the **Your Public Certificates** section, and click the **Add** button. The Add Certificate page appears.
- Click the Browse button, and select the public certificate that you want to upload to PayPal from your local computer (*certs/ewp-cert.pem*).
- Click the **Add** button.
- After your public certificate is uploaded successfully, it appears in the Your Public Certificates section of the Website Payment Certificates page.
- Copy the associated **certificate ID** to *paypal_cfg.php*.

```
define("DEFAULT_EWP_CERT_ID", "certificate_id");
```

4. Downloading the PayPal public certificate from the PayPal website

   - **Log in** to your Business or Premier **PayPal** account.
   - Click the **Profile** subtab.
   - In the Seller Preferences column, click the **Encrypted Payment Settings** link.
   - Scroll down the page to the PayPal Public Certificate section.
   - Click the **Download** button and save the file in *radiusmanager/certs* directory (*paypal-cert.pem*).

Radius Manager binaries

For easier identifying the problems on your system we are describing here the functions of Radius Manager executable files. They are:

1. **rmauth** – Checks for the limits, authenticates users, sets bandwidth etc. It is called from *raddb/users.*
2. **rmacnt** – Closes the inactive accounting sessions. Called from *raddb/acct_users*.
3. **rmpoller** – Checks for the account expiration, disconnects expired users, sends warning emails. It is a standalone daemon process.
4. **rmconntrack** – Handles Mikrotik syslog messages, manages CTS data.
5. **rootexec** – Used to execute external UNIX programs from PHP.
6. **rmscheduler.php** – This program is running regularly from *cron* and it is executed daily once. The recommended time for this is some minutes after midnight. It will check the expired RADIUS accounts, unpaid invoices and disables UNIX users. Also, it is a service type changer for scheduled service changes, disconnects postpaid users on the 1st day of the month (not disables them) for correct postpaid billing and sends warning emails. It is also responsible for account auto renewing.

These binaries store their configuration data in */etc/radiusmanager.cfg* and in *config/ system_cfg.php*.

Radius Manager API

api.php

Name: api_verifyuser

Description: The function is called upon self registering the user, right after submitting the form. From this function You can call your own SMS gateway (HTTP gateway with CURL or a shell script to use your own mobile phone) to send the verification code for the user.

Parameters: $username, $password, $firstname, $lastname, $address, $city, $zip, $country, $state, $phone, $mobile, $email, $srvid, $verifycode, &$errmsg

Results: true - API succeeded
false - API error

Remarks: The function includes an example of integrating the clickatell.com HTTP -> SMS gateway.

LEGAL NOTE