DMA

RADIUS MANAGER

BILLING SYSTEM

INSTALLATION MANUAL

version 4.2

01/30/2018

TABLE OF CONTENTS

FOREWORD

This manual describes the installation procedure for DMA Radius Manager billing system on a CentOS and Ubuntu servers. The following Linux versions are covered:

1. **CentOS 5, 6, 7**
2. **Ubuntu 10–14**

The recommended Linux distributions are **CentOS 5, 6** and **7**, but **Ubuntu 10–14** versions are also usable. **Ubuntu 16** is **not supported** due to incompatibility of **PHP 7**. CentOS is much more flexible than Ubuntu; we strongly recommend CentOS for hosting DMA Radius Manager. The required software packages are available on the installation media and downloadable from the official repositories with **yum** and **apt-get** tools.

This manual covers the DMA Radius Manager **installation** steps for **CentOS 5, 6**, **7** and **Ubuntu 10–14**. You can also find guidelines how to configure RADIUS parameters in the NAS device (Network Access Server) to talk to DMA Radius Manager server.

DMA Radius Manager currently supports the following NAS devices:

1. **Mikrotik 2.8−6.x**. Use final releases only, RC versions are not recommended. The main features are: PPPoE, PPtP, L2tP, Hotspot and Wireless Access List authentication and accounting.
2. **Chillispot** running on Linux server or on DD-WRT device. You can download the tested Linux version from our download portal.
3. **StarOS v2** or **v3** server. Supported features: complete PPPoE and partial RADIUS Wireless Access List support.
4. **Cisco NAS**. Correct IOS version is required. VPDN, BBA GROUP and Virtual template support is necessary to accept RADIUS authenticated PPPoE, PPtP and L2tP calls.
5. **pfSense** Hotspot server.

Radius Manager DOCSIS version supports **cable modem** based Internet distribution systems. With it You can control almost any CMTS device (Cisco, Motorola, Arris etc.) in any mode (routing or bridge). **Date capped** and **uncapped** service plans are supported with **data rate limitation**.

The following steps are necessary to successfully install Radius Manager on a Linux server:

1. **Disable SELinux** (CentOS)
2. Install **ionCube** runtime libraries
3. Build and configure **FreeRadius** server
4. Configure **MySQL** database and credentials
5. Install Radius Manager **WEB** components
6. Install Radius Manager **binaries**
7. Install and configure **DHCP server** (DOCSIS version only)
8. Install **DOCSIS utility** (DOCSIS version only)
9. Complete the **post installation** steps

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

INSTALLATION

Prerequisites

The following components are necessary to successfully install and run the Radius Manager:

Hardware:

- x86 compatible CPU (32 or 64 bit, multi core recommended)
- 1 GB RAM or more (2 GB RAM or more is recommended)
- 80 GB HDD or more (for CTS db 1 TB or more is required)

Software:

- FreeRadius 2.2.0 DMA patch (the latest version is available from www.dmasoftlab.com)
- PHP 5.x (PHP 7 is completely incompatible)
- MySQL 5 or better
- 32 bit glibc
- mysql-devel
- php-mysql
- php-snmp
- php-gd
- php-curl
- php-process
- net-snmp
- net-snmp-utils
- curl
- glibc 2.4 or better
- GNU C/C++ compiler
- DHCP server version 3 (DOCSIS only)
- ionCube runtime libraries
- Javascript enabled WEB browser

Optional components:

- **Webmin** – WEB based Linux configuration tool
- **phpMyAdmin** – WEB based MySQL database frontend
- **Midnight Commander** – An all-in-one system management tool

Preparing the Linux system

CentOS 5, 6, 7

Make sure the required components are available on your Linux server before You proceed the installation of Radius Manager.

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

```
SELINUX=disabled
```

2. Istall the **epel repository**:

```
[root@localhost]# yum install epel-release
```

3. Install the required packages in one step.

CentOS 5:

```
[root@localhost]# yum install mc wget crontabs vixie-cron make gcc libtool-ltdl curl
mysql-server mysql-devel net-snmp net-snmp-utils php53 php53-mysql php53-gd php53-
snmp php53-process ntp sendmail sendmail-cf alpine mutt psmisc
```

CentOS 6:

```
[root@localhost]# yum install mc wget crontabs vixie-cron make gcc libtool-ltdl curl
mysql-server mysql-devel net-snmp net-snmp-utils php php-mysql php-gd php-snmp
php-process ntp sendmail sendmail-cf alpine mutt psmisc
apt-get install mc wget rcconf make gcc mysql-server mysql-client
```

CentOS 7:

```
[root@localhost]# yum install mc wget crontabs make gcc libtool-ltdl curl mariadb-server
mariadb-devel net-snmp net-snmp-utils php php-mysql php-gd php-snmp php-process
ntp sendmail sendmail-cf alpine mutt psmisc net-tools
```

On a 64 bit server install the **32 bit glibc**:

```
[root@localhost]# yum install glibc.i386 libgcc_s.so.1
```

or

```
[root@localhost]# yum install glibc.i686 libgcc_s.so.1
```

Without the 32 bit glibc Radius Manager **binaries** will **not run** (reporting *"no such command is available"* etc., however the executable files are available in */usr/local/bin* directory and file permissions are correct).

Now configure the Linux services.

CentOS 5-6:

```
chkconfig --levels 345 httpd on
chkconfig --levels 345 sshd on
chkconfig --levels 345 mysqld on
chkconfig --levels 345 network on
chkconfig --levels 345 ntpd on
chkconfig --del iptables
service iptables stop
service ntpd restart
```

CentOS 7:

```
systemctl set-default multi-user.target
systemctl enable httpd.service
systemctl enable sshd.service
systemctl enable mariadb.service
systemctl enable ntpd.service
systemctl disable firewalld
systemctl stop firewalld
systemctl restart ntpd.service
```

Ubuntu 10–14

Install the required packages in one step using the command below:

```
[root@localhost]# apt-get install mc wget apache2 make gcc mysql-server mysql-client
libmysqlclient15-dev libperl-dev curl php5 libapache2-mod-php5 php5-mysql php5-cli
php5-curl php5-gd php5-snmp alpine mutt postfix
```

On 64 bit server a 32 bit glibc is also required:

```
[root@localhost]# apt-get install ia32-libs
```

Without the 32 bit glibc Radius Manager **binaries** will **not run** (reporting *“no such command is available”* etc., however the executable files are available in */usr/local/bin* directory and file permissions are correct).

Installation procedure for ionCube runtime system

Radius Manager requires ionCube runtime system. You can download the complete installation package from the address below:

www.dmasoftlab.com/downloads

Before installing ionCube You need to know the following:

1. The **architecture** of your Linux system (32 or 64 bit)
2. The installed **PHP version**
3. The location of **php.ini** file

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 any other file handler.

2. Add the appropriate **ionCube loader** to *php.ini*. For instance, if You have PHP 5.3.3 add the following line:

```
zend_extension=/usr/local/ioncube/ioncube_loader_lin_5.3.so
```

Be sure to enter the correct **PHP version** in the *zend_extension* line. If there are other *zend_extension* entries available in *php.ini*, insert the new *zend_extension* **before** all other existing entries.

On Ubuntu **two php.ini** files can be found:

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

You need to append ionCube lines to **both files**. On CentOS there is only one *php.ini* available (*/etc/php.ini).*

3. **Test** the **ionCube** loader from shell:

```
[root@localhost]# php -v
HP 5.3.3 (cli) (built: Feb  2 2012 23:24:47)
Copyright (c) 1997-2010 The PHP Group
Zend Engine v2.3.0, Copyright (c) 1998-2010 Zend Technologies
    with the ionCube PHP Loader v3.3.14, Copyright (c) 2002-2010, by ionCube Ltd.
```

Assuming You have configured ionCube properly You have to see the correct ionCube version.

4. **Restart** the WEB server (CentOS):

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

Ubuntu:

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

Troubleshooting the ionCube loader system

If ionCube encoded files fail to run You can test the ionCube runtime with *ioncube-loader-helper* file **(**included in the ionCube installation archive).

1. **Copy** *ioncube-encoded-file.php* to **WEB root** directory (on CentOS it is */var/www/html*).

2. Try to **access** the *ioncube-encoded-file.php* script using your WEB browser.

   yourhost/ioncube-encoded-file.php

3. If You see a message "*This file has been successfully decoded. ionCube Loaders are correctly installed*" ionCube is working properly. If You can't decode the file, check *php.ini,* ensure **SELinux** is **disabled** etc. Examine Apache **error log** *(/var/log/httpd/error_log)* for more details.

Notes about PHP safe mode

PHP safe mode (if enabled in *php.ini*) forbids the execution of UNIX commands invoked by Radius Manager via *shell_exec* PHP function. It is recommended to **turn off** PHP **safe mode** to enable all Radius Manager functions. Always check the Apache log if You encounter PHP / Apache related problems (*/var/log* directory).

Installation procedure of FreeRadius

DMA Radius Manager 4.2 requires FreeRadius 2.2.0 DMA patch 2. This custom built FreeRadius version is tested by our software engineers and guarantees 100% compatibility with DMA Radius Manager.

Other versions and builds are incompatible. If your host already has a different FreeRadius installed, remove it completely (delete the */usr/local/etc/raddb* directory completely).

Follow the installation steps below to successfully build, install and configure FreeRadius on your Linux host. All commands should be issued as root user:

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

   www.dmasoftlab.com/downloads

2. Configure and compile **FreeRadius** from sources.

Untar the FreeRadius archive:

```
[root@localhost]# tar xvf freeradius-server-2.2.0-dma-patch-2.tar.gz
```

Prepare the *makefile*:

```
[root@localhost]# cd freeradius-server-2.2.0
[root@localhost]# ./configure
```

Build and install the software:

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

Ensure **mysql-devel** package is installed. By default FreeRadius installs in */usr/local* directory.

3. **Test** FreeRadius in debug mode first. Start it with *radius -X* (upper case 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.
```

You should see "*Ready to process requests*". If *radiusd* cannot find the required libraries, issue *ldconfig* from shell to refresh the **ld** linker cache (required on Ubuntu).

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

If the problem still exists, contact the technical support (support@dmasoftlab.com).

4. If You don’t want to use *install.sh* to install Radius Manager, **set** the correct **owner** of FreeRadius configuration files manually.

On CentOS:

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

On Ubuntu:

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

Radius Manager updates *clients.conf* automatically. It is necessary to set the correct permissions on the affected files.

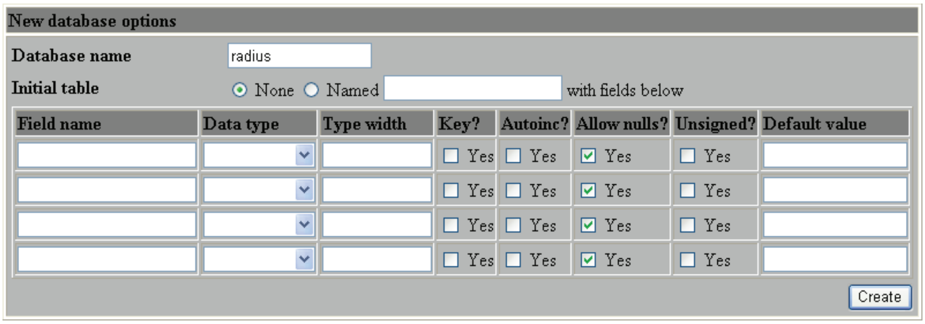
5. **Review** and optionally edit **MySQL credentials** in */usr/local/etc/raddb/sql.conf*:

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

6. Create **MySQL databases** and **MySQL users**. Two methods are described in this manual: **MySQL** command line and **Webmin**.

Preparing MySQL databases with Webmin

Webmin is ideal administration tool for unexperienced Linux users. First, create the RADIUS and CONNTRACK databases. Enter the database name in the right field.



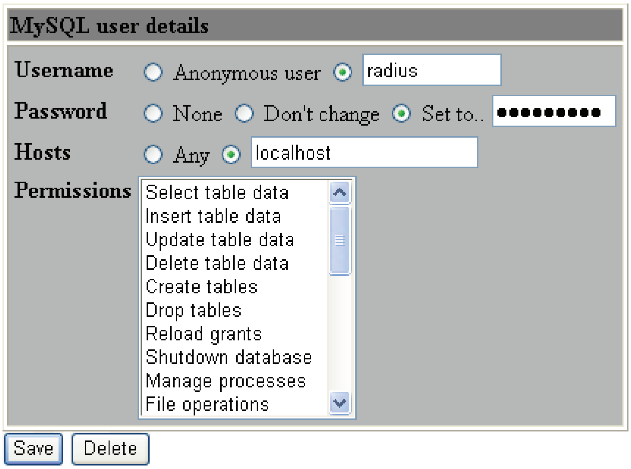
New database options

Database name: radius

Initial table: ● None ○ Named [ ] with fields below

Field name	Data type	Type width	Key?	Autoinc?	Allow nulls?	Unsigned?	Default value
			□ Yes	□ Yes	☑ Yes	□ Yes	
			□ Yes	□ Yes	☑ Yes	□ Yes	
			□ Yes	□ Yes	☑ Yes	□ Yes	
			□ Yes	□ Yes	☑ Yes	□ Yes	

Register **database users**. For default installation set password “**radius123”** for user “**radius”** and “**conn123”** for user **“conntrack”**.



MySQL user details

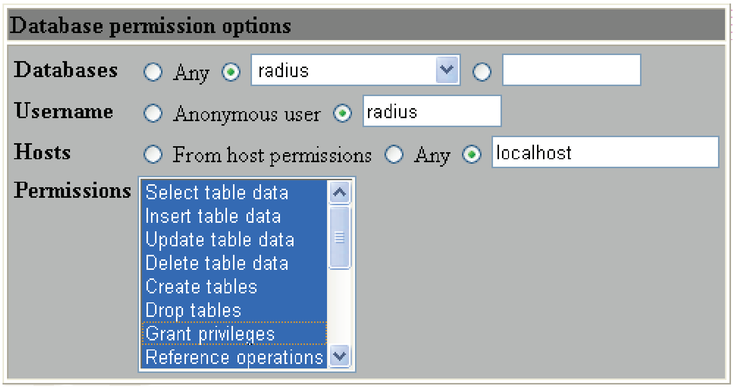
Username: ○ Anonymous user ● radius

Password: ○ None ○ Don't change ● Set to.. ●●●●●●●●●●

Hosts: ○ Any ● localhost

Permissions: Select table data, Insert table data, Update table data, Delete table data, Create tables, Drop tables, Reload grants, Shutdown database, Manage processes, File operations

Set **host permissions**. Select all permissions for both **radius** and **conntrack** users.



Database permission options

Databases: ○ Any ● radius ○

Username: ○ Anonymous user ● radius

Hosts: ○ From host permissions ○ Any ● localhost

Permissions (selected): Select table data, Insert table data, Update table data, Delete table data, Create tables, Drop tables, Grant privileges, Reference operations

Creating MySQL databases with MySQL command line

If You are familiar with MySQL command line, You can create databases, users and permissions in one step.

Log on to MySQL server as root:

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

The *password* is the MySQL root password. If there is no root password set, simply invoke MySQL without any parameters.

Execute the following commands 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;
```

The databases are ready to use.

Installation procedure of DMA Radius Manager

Interactive installation

The easiest way to install DMA Radius Manager is to launch *install.sh* installer script. It is located in Radius Manager tar archive and supports CentOS and Ubntu systems. Before You begin, ensure You have prepared the MySQL database tables and credentials. DMA Radius Manager requires two databases:

1. **RADIUS** – Storage for system data, user base and accounting information.
2. **CONNTRACK** – Connection Tracking System (CTS) storage.

Create **both databases** even if You are not planning to use the CTS module.

After decompressing Radius Manager tar archive (*tar xvf [filename]*), set **755 permission** on *install.sh* and launch it. In the example below we will run *install.sh* on a CentOS 6 system.

```
[root@localhost]# chmod 755 install.sh
[root@localhost]# ./install.sh
Radius Manager installer script
Copyright 2004-2018, DMA Softlab LLC
All right reserved.

(Use CTRL+C to abort any time)

Select the type of your operating system:
1. CentOS 5-6-7
2. Ubuntu 10-13
3. Ubuntu 14

Choose an option: [1]
```

Select the correct operating system You have.

Next, select the installation method:

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

Choose an option: [1]
```

Select option **1** for new installation. The default option is displayed after each question. You can just press enter in most cases.

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

Enter the full path of **HTTP root directory**. The installer will create *radiusmanager* subdirectory in it. On CentOS simply press enter.

Enter the MySQL database credentials as You defined them beforehand:

```
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 default setup simply press enter to use MySQL user “**radius**” / “**radius123**” for the **RADIUS** database and “**conntrack**” / “**conn123**” for the **CONNTRACK** database. The default database host is “**localhost**”. Enter custom values if You have a different setup,

It is strongly recommended to configure a separate database host for CONNTRACK database If You are planning to control hundreds of online users (> 500).

Next step is to enter the FreeRadius user name. It is required to set the correct permission on /etc/*radiusmanager.cfg*. Radius Manager binaries will not run if there is a permission problem.

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

On CentOS and Ubuntu the FreeRadius user name is **root**.

Now enter the **Apache user** name. It is required to set the correct permission on files in *radiusmanager/* directory. On CentOS it is **apache**, while on Ubuntu it is **www-data**.

```
HTTPD UNIX user: [apache]
```

Now You are asked to register **rmpoller service**. It is a standard CentOS / Ubuntu compatible service which starts *rmpoller* at system boot.

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

In most cases You can simply press enter. When the service has been created, You can use the CentOS command

service rmpoller [start | stop]

to control the **rmpoller** service activity. Make this service auto starting at boot time together with FreeRadius. Use *chkconfig* command (CentOS) or Webmin to activate the service at boot time. Rmpoller must be **running all time**.

Select ‘**y**’ if You want to register the **rmconntrack** service. It is a standard Linux service and required by the **CTS** module.

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

Once the service has been registered, You can use the command

service rmconntrack [start | stop]

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

It is strongly recommended to back up the complete RADIUS database before You continue the installation. Answer ‘**y**’ to the following question:

```
Back up RADIUS database: [y]
```

The installer answers with

```
WARNING! If You continue the existing RADIUS database will be overwritten!

Are You sure to begin the installation? [n]
```

Press ‘**y**’ to continue or ‘**n**’ to abort the process. You can press **Ctrl+C** any time to abort the installation.

```
Starting installation...

Stopping rmpoller
Stopping rmconntrack
Stopping radiusd
Stopping rmauth
Stopping rmacnt
Backing up radiusmanager.cfg
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
Backing up RADIUS database...
Creating MySQL tables
Enabling rmpoller service at boot time
Enabling rmconntrack service at boot time
Enabling radiusd service at boot time
Copying logrotate script
Copying cronjob script
Setting permission on raddb files

--------------------LICENSE REQUEST BEGIN---------------------
NYNewe7RFgGzcYVsKTcN7cuklOYqIULMDabgnN33UXi4JEwhifm1WBK/3W4U22OA
SINqBLnM5s+8AofF/PVEiPGB5ZfxsnOfMuDMw5Q9aV+uARogaBhiL9LlSOSRVNoS
EiLKaqdPPgfzhIWlOA7Jg8YWlClha6Gu9WQG6OLzzOdNYDHnaScdeiOswfHiiXsD
3hPFtiCrYCGh3PQboUDqJmvYBKfle/rxTH61g6kCMuZ2Cu0DmfUf3c//HDMih0Fv
lPouhyIWKsxIkrlBef73+HPkn6G0yaIzFUmXcl7uvKecHOKoudtCT110eREJQPxc
G/ZcPpccVAUzMzgkaCwI/Q==
---------------------LICENSE REQUEST END----------------------

Installation complete!
```

At this step log into the DMA customer portal (customers.dmasoftlab.com) and **request a license key**. Enter the **MAC address** of the NIC (use *ifconfig* command to find the MAC address of the ethernet card) and the **license key request code**.

DMA Radius Manager **will run** on a **licensed host** only. The license is bound to various hardware and OS software components. Licensing policy is available in **Terms and Conditions** on DMA Softlab website.

Once the license key is issued, icopy the *lic.txt* and *mod.txt* to radiusmanager WEB directory. Try to **access the ACP** (Administration Control Panel). **Reboot** your system to check if all services are started properly (*radiusd*, *rmpoller* and optionally *rmconntrack*)?

Launch *radiusd* 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.
```

Issue the following command in the second terminal window:

```
[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 any error, check the following:

- Is MySQL server running?
- Are MySQL credentials correct?
- Are MySQL table permissions correct?
- Can FreeRadius connect to MySQL database?
- Are RADIUS and CONNTRACK databases, tables available?
- Is the NAS defined in ACP? In this example the NAS IP address is 127.0.0.1.
- Is the hostname available in */etc/hosts* file?
- Sometimes it is necessary to define the real IP of Linux in RM ACP / Host list (for *radtest* testing only).

You can examine the detailed error message in **radiusd -X** debug output. First, stop the running daemon:

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

or

```
[root@localhost]# ps ax | grep radius
[root@localhost]# kill [pid]
```

Substitute the PID with the correct PID (process id). Now activate the debug mode:

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

Run **radtest** or try to authenticate users on a real NAS. In the debug output You will see the correct *NAS-IP-Address* what You need to enter in Radius Manager ACP / NAS list.

If there are errors like *“Ignoring request from unknow NAS”* or *“NAS not found”*, the NAS is not defined in ACP. Stop the *radius* process (CTRL + C), enter the correct NAS IP address in ACP and restart debug mode with *radiusd -X.* You can use the same method every time if a new NAS won’t work.

Beginning from Radius Manager v 4.1 *radiusd* is **restarting automatically** upon updating any NAS in ACP.

Manual installation

The method below describes the manual installations steps for DMA Radius Manager. It is intended to use by professional system administrators who want to control every step of the installation.

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

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

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

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

4. Review and optionally customize /etc/*radiusmanager.cfg*.

5. Change the **permission** and **owner** on /etc/*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 correct user. It must be the FreeRadius user (**root** in most cases), otherwise the binaries will not be able to read the configuration file.

6. Test **rmauth** from shell:

```
[root@localhost]# rmauth -v
rmauth version 4.2.0, build 4558 (20180120)
Copyright 2004-2018, DMA Softlab
All rights reserved.
```

You have to see similar result as shown above. If there are errors, maybe You have an old glibc package or some libraries are missing. In this case try to install the missing packages. It You can't fix it, contact the DMA Softlab technical support (support@dmasoftlab.com).

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 as shown above. If there is a MySQL socket error, enter the correct socket location in */etc/radiusmanager.cfg*. The default socket on **CetnOS** is */var/lib/mysql/mysql.sock*, while on **Ubuntu** it is */var/run/mysqld/mysqld.sock*.

You have to register the NAS entries in ACP to successfully test *rmauth*. In this example the NAS IP address 192.168.0.8 has already been entered in Radius Manager ACP and Mikrotik NAS type has been selected.

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

8. Change rootexec **permission** 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 is password protected.

9. Copy the **radiusmanager** cron file to */etc/cron.d* and set the correct permission:

```
[root@localhost]# chmod 644 /etc/cron.d/radiusmanager
```

10. **Copy** the complete Radius Manager WEB content to **Apache root** directory.

11. **Protect** the configuration files in *radiusmanager/config* directory to be readable by **root** and **Apache** (on Ubuntu it is the **www-data** user):

```
[root@localhost]# cd /var/www/html/radiusmanager/config

[root@localhost]# chown apache 2co_cfg.php authorizenet_cfg.php citrus_cfg.php dps_cfg.php payfast_cfg.php payfast_cfg.php paypal_cfg.php sagepay_cfg.php system_cfg.php

[root@localhost]# chmod 600 2co_cfg.php authorizenet_cfg.php citrus_cfg.php dps_cfg.php payfast_cfg.php payfast_cfg.php paypal_cfg.php sagepay_cfg.php system_cfg.php
```

12. Set the correct owner on **tmpimages** directory. Without this step the online user list will report "*Unable to create image*".

On CentOS:

```
[root@localhost]# chown apache /var/www/html/radiusmanager/tmpimages
```

On Ubuntu:

```
[root@localhost]# chown www-data /var/www/radiusmanager/tmpimages
```

13. **Edit** *system_cfg.php* and review all other configuration files in *config* directory. Read the **Reference** chapter for details.

14. **Install** the initial database **tables.** Execute the next commands:

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

15. Launch a 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 test the menu functions.

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

http://yourhost/radiusmanager/user.php

The initial username and password are:

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

MySQL optimization

The performance of Radius Manager system depends mainly on the speed of hard disk and MySQL server. Correct InnoDB configuration is required to achieve good RADIUS response time.

1. Check the **radacct** table **size**. If it is larger than 3-4 GB, prune the accounting table with *dbcleanup.sql* script (included in *SQL* directory).
2. **Add** more **RAM** to system. Adding 2-4 GB RAM doesn't mean any problem nowadays.
3. Use **RAID 0**, **1** or **5** array as MySQL storage device. Hardware RAID controller is recommended.
4. **Optimize** the **MySQL** in *my.cnf*

Add the following entries to */etc/my.cnf* in *mysqld* section:

```
innodb_buffer_pool_size=512M
innodb_log_file_size=128M
innodb_file_per_table
innodb_flush_log_at_trx_commit=2
innodb_flush_method=O_DIRECT
```

Set **innodb_buffer_pool_size = 50%** of RAM size and **innodb_log_file_size = 128**. The configuration snippet above is for a system with 1 GB RAM. For 2 GB RAM or more set **innodb_log_file_size = 256 MB.**

Delete **ib_logfile0** and **ib_logfile1** files in */var/lib/mysql* directory and **restart MySQL** server.

Adding more RAM will drastically speed up the MySQL operations. Indexes should fit in the RAM for optimal performance.

Notes

By default the WEB server lists the contents of the directory where Radius Manager files are stored. With a *.htaccess* file this can be avoided easily. Enable the **Options -Indexes** directive in *.htaccess* file. Enable **htaccess support** in order to use *.htaccess* files (set **AllowOverride All** directive in *httpd.conf* – CentOS). Radius Manager installs a preconfigured *.htaccess* file.

On Ubuntu 14 the following directive is required in */etc/apache2/sites-enabled/000-default.conf* to enable the htaccess support:

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

<Directory /var/www/html>
AllowOverride All
</Directory>
```

SOFTWARE UPGRADE

The following upgrade modes are available:

1. **Interactive**
2. **Manual**

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

Upgrading FreeRadius

Radius Manager requires the latest FreeRadius 2.2.0 DMA patch. Remove any old versions and install the correct FreeRadius on your host. Consult the FreeRadius installation chapter of this manual for details.

Before You proceed the installation of the new FreeRadius, **rename** the **raddb** directory to **raddb.bak** to force FreeRadius to install the new configuration files. Without this step the old, incompatible configuration files will **remain unchanged**.

Configure files in **raddb** directory as it is described in the FreeRadius installation chapter. Do not forget to set the proper **permission** on **raddb** files.

Optimizing MySQL for InnoDB

Radius Manager v 4.0.0 and later versions use InnoDB tables instead of MyISAM. InnoDB is faster, uses row level locking mechanism etc. Radius Manager is more responsive with InnoDB.

Before beginning the upgrade it is important to **optimize** the **MySQL** database engine. Add the following entries to */etc/my.cnf* in *mysqld* section:

```
innodb_buffer_pool_size=512M
innodb_log_file_size=128M
innodb_file_per_table
innodb_flush_log_at_trx_commit=2
innodb_flush_method=O_DIRECT
```

Set **innodb_buffer_pool_size = 50%** of RAM size and **innodb_log_file_size = 128**. The configuration snippet above is for a system with 1 GB RAM. For 2 GB RAM or more set **innodb_log_file_size = 256 MB.**

Delete **ib_logfile0** and **ib_logfile1** files in */var/lib/mysql* directory and **restart MySQL** server.

Adding more RAM will drastically speed up the MySQL operations. Indexes should fit in the RAM for optimal performance.

Without these setting the upgrade procedure can last several hours and the overall system performance will be poor.

Interactive upgrade

Radius Manager installer script can upgrade the installed system automatically. Complete the following steps as explained below.

Decompress Radius Manager tar archive.

```
[root@localhost]# tar xvf radiusmanager-4.2.0.tgz
```

Go to *radiusmanager* directory and set **755 permission** on *install.sh*.

```
[root@localhost]# cd radiusmanager
[root@localhost]# chmod 755 install.sh
```

Launch *install.sh* and select your **Linux version**:

```
[root@localhost]# chmod 755 install.sh
[root@localhost]# ./install.sh
Radius Manager installer script
Copyright 2004-2018, DMA Softlab LLC
All right reserved.

(Use CTRL+C to abort any time)

Select the type of your operating system:
1. CentOS 5-6-7
2. Ubuntu 10-13
3. Ubuntu 14

Choose an option: [1]
```

Select option **2** for upgrade:

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

Choose an option: [1]
```

Choose the **currently installed** Radius Manager version.

WARNING! 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
21. v3.8.0
22. v3.9.0
23. v4.0.x
24. v4.1.x

Select current installed version: 23
```

Enter the location of the **HTTP root** directory:

```
Current installed version is 4.1.x
WWW root path: [/var/www/html]
Directory /var/www/html/radiusmanager already exists. Overwrite? [n]
```

The installer will ask You to allow overwriting existing files in *radiusmanager* directory. Answer ‘**y**’. The installer will back up the configuration files in *config* directory. Do not reuse the old format configuration files, customize the newly installed ones.

Now enter 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]
```

For default setup simply press enter to use MySQL user “**radius**” / “**radius123**” for the **RADIUS** database and “**conntrack**” / “**conn123**” for the **CONNTRACK** database. The default database host is “**localhost**”. Enter custom values if You have a different setup,

Next step is to enter the FreeRadius user name. It is required to set the correct permission on /etc/*radiusmanager.cfg*. Radius Manager binaries will not run if there is a permission problem.

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

On CentOS and Ubuntu the FreeRadius user name is **root**.

Now enter the **Apache user** name. It is required to set the correct permission on files in *radiusmanager/* directory. On CentOS it is **apache**, while on Ubuntu it is **www-data**.

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

Now You are asked to register **rmpoller service**. It is a standard CentOS / Ubuntu compatible service which starts *rmpoller* at system boot.

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

In most cases You can simply press enter. When the service has been created, You can use the CentOS command

service rmpoller [start | stop]

to control the **rmpoller** service activity. Make this service auto starting at boot time together with FreeRadius. Use *chkconfig* command (CentOS) or Webmin to activate the service at boot time. Rmpoller must be **running all time**.

Select ‘**y**’ if You want to register the **rmconntrack** service. It is a standard Linux service and required by the **CTS** module.

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

Once the service has been registered, You can use the command

service rmconntrack [start | stop]

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

It is strongly recommended to back up the complete RADIUS database before You continue the installation. Answer ‘**y**’ to the following question:

```
Back up RADIUS database: [y]
```

The installer answers with

```
WARNING! Back up the complete RADIUS database before You proceed!

Are You sure to begin the upgrade? [n]
```

IMPORTANT! Back up the **complete database** at this point!

Press ‘**y**’ to continue or ‘**n**’ to abort the process. You can press Ctrl+C any time to abort the installation.

```
Starting installation...

Stopping rmpoller
Stopping rmconntrack
Stopping radiusd
Stopping rmauth
Stopping rmacnt
Backing up radiusmanager.cfg
Backing up system_cfg.php
Backing up sagepay_cfg.php
Backing up paypal_cfg.php
Backing up authorizenet_cfg.php
Backing up dps_cfg.php
Backing up 2co_cfg.php
Backing up payfast_cfg.php
Backing up citrus_cfg.php
Backing up smsgateway.php
Backing up dhcpd.conf
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
Backing up RADIUS database...
Upgrading MySQL tables. Please be patient.
Upgrading to version 4.2
ERROR 1054 (42S22) at line 1: Unknown column ‘enableapi’ in ‘nas’
Enabling rmpoller service at boot time
Enabling rmconntrack service at boot time
Enabling radiusd service at boot time
Copying logrotate script
Copying cronjob script
Setting permission on raddb files

--------------------LICENSE REQUEST BEGIN---------------------
nc3RMLwzO4VzVkdI4BHWWIbY2ZIYhD4/LUPqgItGNjWha7BnhTb2NG3taQc5cDw5
Yr9orE39OXb8KcmfEtqPO3o8ywfDUyRHhBqBgOLsNSCiHbdbXxYMbNubFSQqikaH
DT8aV6KYRI6rgO4DY9DwgYL6rzJ06bxV3zSzzXbQPIL8ctdvBYMxsmbqyjBHjSbR
HR1uZgkbTjC+F9oaksACxI3NKYqR03ZAZtEcJFSFX0h8TajeTkrtj1fzotZidIQD
OwFsUfOmfxr1+1MZAjF8NtrKgUfutehJQcEURhK9ZK7Ui07fltV81KRs8jXPVQKz
+6am3kz5XtyyiR0L4Ahh9w==
---------------------LICENSE REQUEST END----------------------

Installation complete!
```

No error message should displayed during the upgrade.

Manual upgrade

In manual upgrade mode You have to check / reinstall / reconfigure the following components:

1. Upgrade **FreeRadius**
2. Upgrade Radius Manager **binaries**
3. **Optimize MySQL** server (*my.cnf*)
4. Upgrade RADIUS **database**
5. Upgrade Radius Manager **WEB components**
6. Configure **cron**

Updating FreeRadius

DMA Radius Manager 4.2 requires FreeRadius 2.2.0 DMA patch 2. Find the FreeRadius installation procedure in “Installation procedure of FreeRadius” chapter of this manual.

Updating Radius Manager executables

Install the new **rmauth**, **rmacnt**, **rmpoller**, **rmconntrack** and **rootexec** executables. Follow paragraphs 1–12 from “Manual installation” chapter. **Stop rmpoller** and **rmconntrack** daemons before You can upgrade them. Issue the following commands (CentOS):

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

On other systems use the following method. Enter the correct PID in **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
```

Optimizing MySQL

Before beginning the upgrade it is required to **optimize MySQL server**.

Add the following entries to */etc/my.cnf* in *mysqld* section:

```
innodb_buffer_pool_size=512M
innodb_log_file_size=128M
innodb_file_per_table
innodb_flush_log_at_trx_commit=2
innodb_flush_method=O_DIRECT
```

Set **innodb_buffer_pool_size = 50%** of RAM size and **innodb_log_file_size = 128**. The configuration snippet above is for a system with 1 GB RAM. For 2 GB RAM or more set **innodb_log_file_size = 256 MB.**

Delete **ib_logfile0** and **ib_logfile1** files in */var/lib/mysql* directory and **restart MySQL** server.

Adding more RAM will drastically speed up the MySQL operations. Indexes should fit in the RAM for optimal performance.

Without this optimization the upgrade procedure can last several hours and the overall system performance will be poor.

Upgrading MySQL tables

To upgrade from an older Radius Manager version to the latest You need to execute **multiple SQL** scripts in **correct order**. For example if You are upgrading Radius Manager from 3.7.0 to 4.2 You have to execute the following SQL scripts (RADIUS db):

1. upgrade-3.7.0_3.8.0.sql
2. upgrade-3.8.0_3.9.0.sql
3. upgrade-3.9.0_4.0.0.sql
4. upgrade-4.0_4.1.sql
5. upgrade-4.1_4.1.sql

To upgrade the CONNTRACK database execute the following scripts in the **correct order**:

1. upgrade_cts-3.7.0_3.8.0.sql
2. upgrade_cts-3.8.0_3.9.0.sql
3. upgrade_cts-3.9.0_4.0.0.sql
4. upgrade_cts-4.0_4.1.sql
5. upgrade_cts-4.1_4.2.sql

Installing new PHP files

Copy the complete *radiusmanager* WEB directory, overwriting the old files. Be sure to back up the old configuration files before overwriting them. When done, review and modify the new configuration files. The configuration files are changing from version to version; You have to edit them every time after updating the system. **Do not use the old configuration files!** They are incompatible.

Copy the **radiusmanager** cron file to */etc/cron.d* and set the correct permission:

```
[root@localhost]# chmod 644 /etc/cron.d/radiusmanager
```

Set the **permissions** and **ownership** on all **PHP files** as described in the manual installation chapter.

Cron

Radius Manager 4 and newer versions use a separate **crontab** file. It is neccessary to **remove** *rmscheduler.php* from */etc/crontab*. Open */etc/crontab* in any text editor and delete the *rmscheduler .php* line.

Install **radiusmanager** in */etc/cron* directory.

WARNING

- When upgrading to 3.0.0 the **invoice sum** and **payout** data are **lost** due to the new data storage mechanism.
- **Back up** the complete **database** before the upgrade!
- When upgrading to 3.8.0 the old **invoice sums** can be **wrong** due to new structure of rm_invoices table. If You have not printed the old invoices yet, do it before upgrading to 3.8.0.

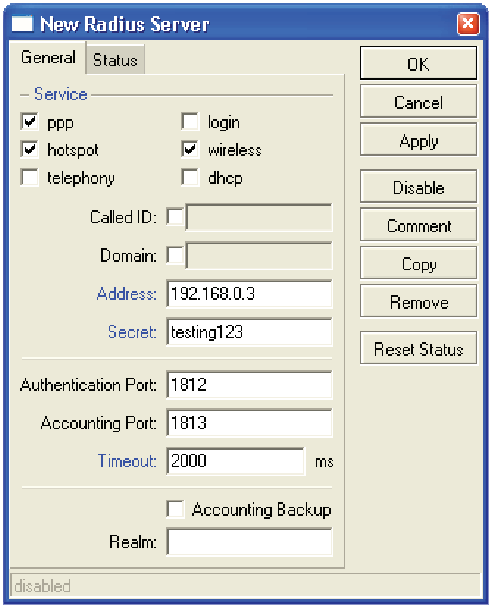
NAS CONFIGURATION

Mikrotik

Enabling RADIUS authentication and accounting

You have to configure the Mikrotik NAS to forward the authentication and accounting requests to RADIUS server. Use Winbox to view and edit the configuration. Follow the steps below:

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



Service
☑ ppp ☐ login
☑ hotspot ☑ wireless
☐ telephony ☐ dhcp

Called ID: ☐

Domain: ☐

Address: 192.168.0.3

Secret: testing123

Authentication Port: 1812

Accounting Port: 1813

Timeout: 2000 ms

☐ Accounting Backup

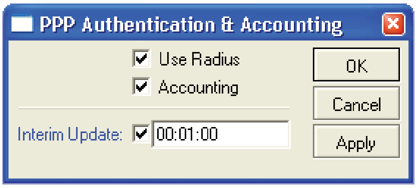
Realm:

disabled

Options are:

- **Service**:
  - **Hotspot**: enable Hotspot RADIUS authentication.
  - **Wireless**: enable Wireless Access List RADIUS authentication (uncheck *Default authenticate* in WLAN settings and enable RADIUS MAC authentication in the selected *security profile*)
  - **PPP**: PPP RADIUS authentication (PPPoE, PPtP, L2tP).
  - **Login**: Winbox (Telnet, SSH) authentication with RADIUS.
  - **Telephony**: telephony authentication with RADIUS.
- **Address** is the IP address of your RADIUS server.
- **Secret** is the NAS secret as defined in *ACP / Edit NAS* form.
- **Authentication and Accounting** ports are the standard RADIUS ports (1812, 1813).
- **Timeout:** How many ms to wait for the RADIUS response. If the latency time of RADIUS server is high or the RADIUS accounting table is very large, set this timeout to a higher value (3000-5000 ms). The recommended value is 2000 ms.

4. Set the **AAA options** for **PPP** service (PPtP, L2tP or PPPoE):



PPP Authentication & Accounting

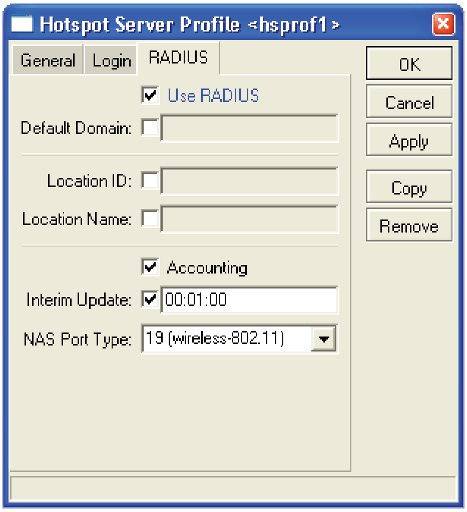
☑ Use Radius

☑ Accounting

Interim Update: ☑ 00:01:00

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 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 <hsprof1>

☑ Use RADIUS

Default Domain: □

Location ID: □

Location Name: □

☑ Accounting

Interim Update: ☑ 00:01:00

NAS Port Type: 19 (wireless-802.11)

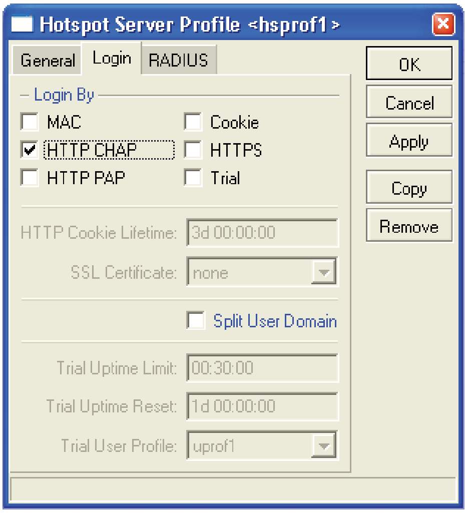
Options are:

- **Use RADIUS** – Enable RADIUS Hotspot authentication.
- **Accounting** – Enable RADIUS Hotspot accounting.
- **Interim update** – Set the interval when RADIUS accounting information is periodically refreshed. Enter 1-5 minutes here. Lower values generate heavy load on MySQL server.

Configure the Hotspot **Login by** options:

- **MAC** – Hotspot MAC authentication method.
- **HTTP CHAP** – Enable HTTP CHAP authentication method. CHAP uses encrypted packets to send the username / password to RADIUS. Always use CHAP if the browsers support it.
- **HTTP PAP** – Enable HTTP PAP authentication method. It has no encryption and can be used as fallback option.
- **Cookie** – If checked the Hotspot login page will remember the username and password.
- **HTTP cookie lifetime** – Defines how many days to remember the username and password.

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



Hotspot Server Profile <hsprof1>

Login By

☐ MAC ☐ Cookie

☑ HTTP CHAP ☐ HTTPS

☐ HTTP PAP ☐ Trial

HTTP Cookie Lifetime: 3d 00:00:00

SSL Certificate: none

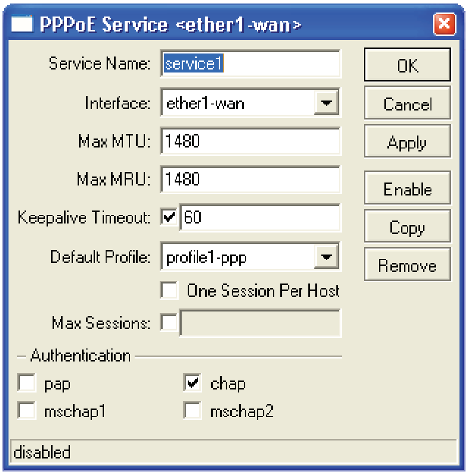
☐ Split User Domain

Trial Uptime Limit: 00:30:00

Trial Uptime Reset: 1d 00:00:00

Trial User Profile: uprof1

Enter the following data:



PPPoE Service <ether1-wan>

Service Name: service1

Interface: ether1-wan

Max MTU: 1480

Max MRU: 1480

Keepalive Timeout: ☑ 60

Default Profile: profile1-ppp

☐ One Session Per Host

Max Sessions: ☐

Authentication

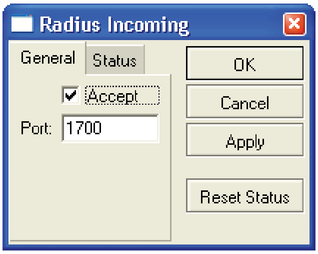
☐ pap ☑ chap

☐ mschap1 ☐ mschap2

disabled

- **Service name –** Service name for PPPoE dialer.
- **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, e.g. 1400).
- **PAP** or **CHAP** authentication method. **CHAP** is recommended, don't enable MSCHAP1 and MSCHAP2. PAP can be used as fallback.
- **Default profile** – Select your PPP profile.
- **Keepalive timeout** – Enter 30-60 seconds here.

7. Enable **incoming RADIUS** requests (POD packets). It is required to enable the REMOTE disconnection method in Radius Manager.



☑ Accept

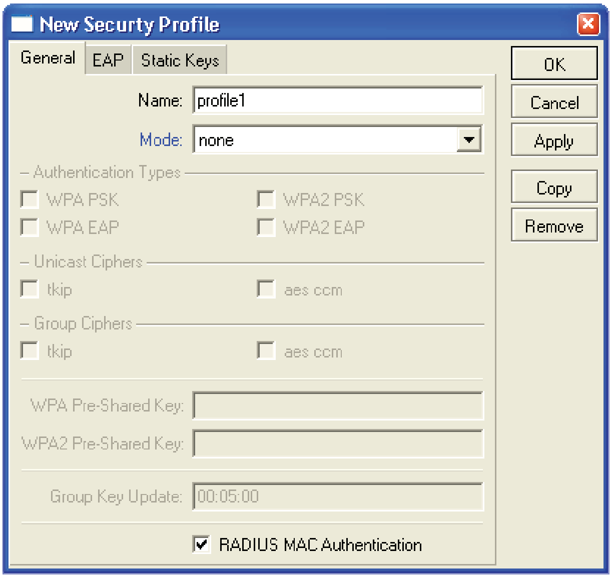
Port: 1700

Don’t forget to open UDP port 1700 in firewall.

RADIUS Access List support (RADIUS ACL)

By default all wireless clients can connect to your Mikrotik wireless AP. You can enable **RADIUS Access List** support if You want to filter the CPE devices and allow only registered clients to connect to an SSID.

1. Register a new **security profile**:



Name: profile1

Mode: none

Authentication Types

□ WPA PSK □ WPA2 PSK

□ WPA EAP □ WPA2 EAP

Unicast Ciphers

□ tkip □ aes ccm

Group Ciphers

□ tkip □ aes ccm

WPA Pre-Shared Key:

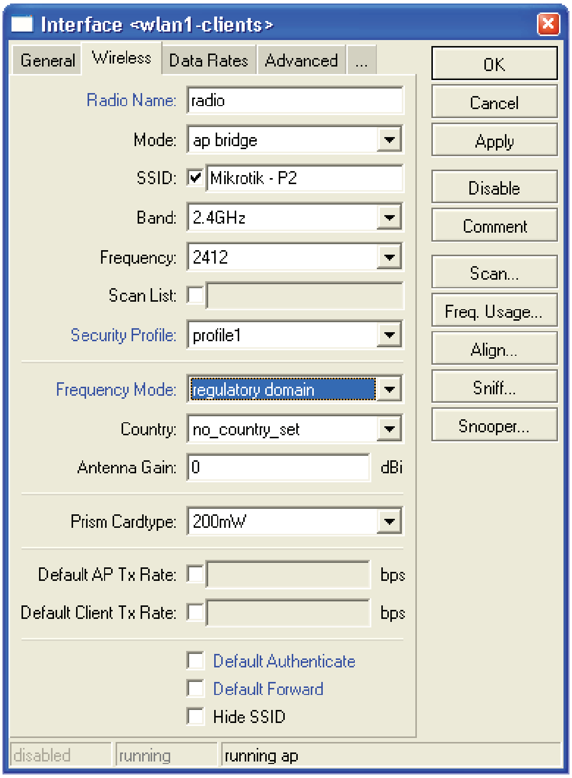
WPA2 Pre-Shared Key:

Group Key Update: 00:05:00

☑ RADIUS MAC Authentication

Check the **RADIUS MAC Authentication** checkbox.

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



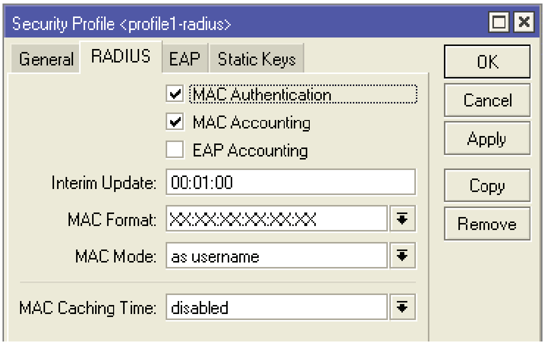
Radio Name: radio
Mode: ap bridge
SSID: ☑ Mikrotik - P2
Band: 2.4GHz
Frequency: 2412
Scan List: □
Security Profile: profile1
Frequency Mode: regulatory domain
Country: no_country_set
Antenna Gain: 0 dBi
Prism Cardtype: 200mW
Default AP Tx Rate: □ bps
Default Client Tx Rate: □ bps
□ Default Authenticate
□ Default Forward
□ Hide SSID
disabled | running | running ap

When a client tries to connect to SSID Mikrotik will authenticate the client's MAC address using the RADIUS server. If the MAC can be found in the database, Mikrotik will allow the connection.

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

MAC authentication and accounting

Wireless MAC authentication / accounting is also available with some limitations. This authentication method doesn't support **data rate** selection.



☑ MAC Authentication
☑ MAC Accounting
□ EAP Accounting
Interim Update: 00:01:00
MAC Format: XX:XX:XX:XX:XX:XX
MAC Mode: as username
MAC Caching Time: disabled

Complete the following steps to enable wireless MAC RADIUS authentication on a Mikrotik NAS:

1. Register a new **wireless security profile** in Mikrotik. In RADIUS tab check **MAC authentication** and **MAC accounting** checkboxes. Set the **interim update** value (1-5 minutes).
2. Select the new security profile in Wireless tab of WLAN card.
3. Enable **Wireless authentication** in Mikrotik RADIUS profile.
4. Register **MAC accounts** in ACP.

The MAC format should be set to **xx:xx:xx:xx:xx:xx**. Select “**as username**” in MAC mode list.

If there are authentication issues You can run *radiusd -X* command to examine the RADIUS log and fix the problem.

Chillispot

Radius Manager supports various Chillispot systems:

1. **Chillispot 1.1.0 Linux** version. It is available from www.dmasoftlab.com.
2. Chillispot running on **DD-WRT** router.
3. Chillispot running on **other router**.

Radius Manager requires properly configured Chillispot server. You have to set **radiuslisten** and **coaport** directives properly.

Chillispot on Linux

You can build Chillispot from sources easily. The following hardware and software components are required to successfully install and configure Chillispot on a Linux server:

- **CentOS** Linux server
- **Two Ethernet** interfaces (for Internet connection and for Hotspot clients)
- **C/C++** development system

1. **Download** the Chillispot source archive and **decompress** it:

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

2. Go to Chillispot directory and prepare the **Makefile**:

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

3. **Build** and **install** Chillispot:

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

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

Now You can test the Chillispot executable with the following command:

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

If You get an error like

“chillispot[8792]: chilli.c: 917: radiussecret must be specified”

it is absolutely normal. You have to edit */etc/chilli.conf* first.

5. Uncomment **debug flags** in line 9:

```
fg
```

Uncommenting this line enables Chillispot to run in foreground mode. It is required for debugging. When the system is fully working, You can comment out the line again to enable the daemon mode.

6. Enter the **DNS** server IP address in line 59:

```
dns1 192.168.0.3
```

It should be a valid, reachable DNS server, otherwise clients will unable to access even the login page. Install and configure **Bind** on your Linux host and enter the IP address of Linux as DNS server.

7. Enter **RADIUS server** addresses in lines 113 and 120:

```
radiusserver1 192.168.0.3
radiusserver2 192.168.0.3
```

It is the address of Radius Manager server. Enable only one server. Enter the same IP address twice.

You can install **FreeRadius**, **Radius Manager** and **Chillispot** on a **same host**, but multiple host installation is also supported.

8. Uncomment line 139 and enter the **RADIUS secret**:

```
radiussecret testing123
```

The secret key should match what is defined in ACP / Edit NAS form.

9. Define RADIUS **NAS IP** in line 149. It is important to send the correct NAS IP in every RADIUS package for correct NAS identification.

```
radiusnasip 192.168.0.3
```

10. Define **UAM** server in line 237:

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

The default gateway address is 192.168.182.1. A HTTPS capable WEB server is required to serve the CGI version of Chillispot login page.

11. **Uncomment** line 248 and define the UAM secret:

```
uamsecret secret
```

This secret should be the same which is defined in *hotspotlogin.cgi*.

11. **Copy** *hotspotlogin.cgi* to *cgi-bin* folder. On CentOS it is */var/www/cgi-bin*. The file *hotspotlogin.cgi* must be executable: set the correct **permissions** using *chmod*:

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

Completing this step Chillispot is redy to use. Now You have to set up a dedicated Ethernet interface in Linux server for Hotspot users. You need two network interface cards (NIC) in your host:

1. **WAN** – for connecting to the Internet.
2. **LAN** – for connecting Chillispot Hotspot clients.

The Hotspot interface (LAN) requires a special setup:

1. **Turn off** all **DHCP** servers if running.
2. **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 on interface eth1. Create a similar *ifcfg-xxx* file and restart the network with **service network restart** command.

If You execute *ifconfig* command You have to see similar results 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 testing the Chillispot. Start it with the following parameters:

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

The parameter *--coaport* defines the port for the incoming disconnect requests (POD). Use value 3779.

After Chillispot has been started, the connected CPE device has to get an IP address from the 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.

IP forwarding and masquerading should be enabled on the Linux host. You can do this with the following command:

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

Masquerade the local Hotspot addresses:

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

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 NAS and begin using your newly installed Chillispot Hotspot system.

DD-WRT

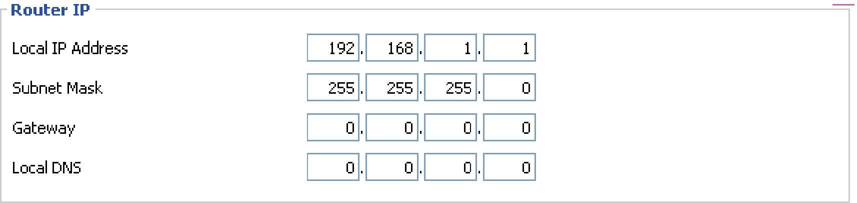
Radius Manager supports authentication and accounting on DD-WRT routers. The following setup instructions are for DD-WRT v2.3 SP3, but You can use it for configuring any other DD-WRT versions (consult your DD-WRT manual first).

As a first step You have to configure the network interfaces on DD-WRT router:

1. **WAN** – Internet side.
2. **LAN & WLAN** – Client side.

WAN is used to connect the router to the Internet. Several connection modes are available. In this example we'll use static IP mode with address 192.168.0.50. You can also enable PPP and DHCP mode on the WAN interface. Set the IP address, netmask, DNS and gateway.

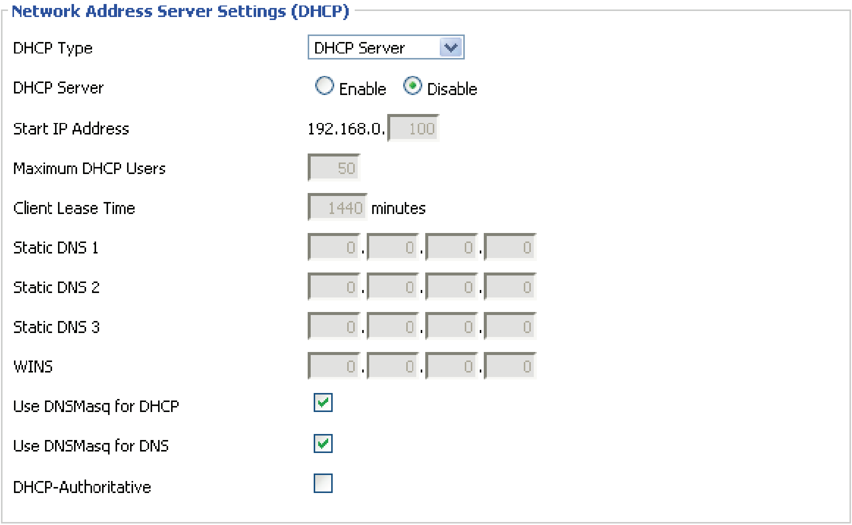
Also set the IP address of the LAN adapter:



Router IP

Local IP Address	192 . 168 . 1 . 1
Subnet Mask	255 . 255 . 255 . 0
Gateway	0 . 0 . 0 . 0
Local DNS	0 . 0 . 0 . 0

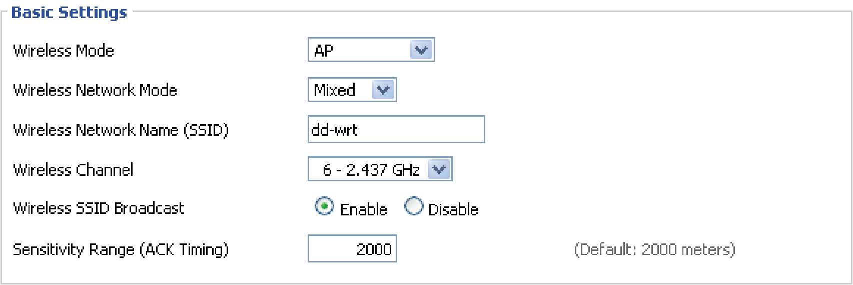
Disable the DHCP server on LAN. Chillispot itself is a DHCP server. A second DHCP server on the same interface will conflict.



Network Address Server Settings (DHCP)

DHCP Type	DHCP Server
DHCP Server	○ Enable ● Disable
Start IP Address	192.168.0. 100
Maximum DHCP Users	50
Client Lease Time	1440 minutes
Static DNS 1	0 . 0 . 0 . 0
Static DNS 2	0 . 0 . 0 . 0
Static DNS 3	0 . 0 . 0 . 0
WINS	0 . 0 . 0 . 0
Use DNSMasq for DHCP	☑
Use DNSMasq for DNS	☑
DHCP-Authoritative	□

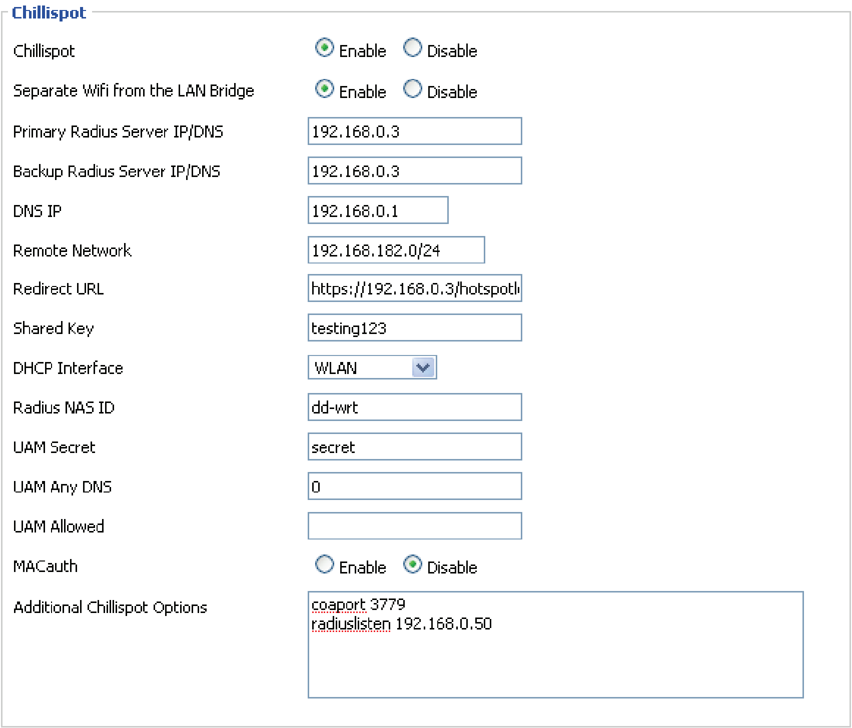
Activate the WLAN interface, enable AP mode, set SSID and channel.



Basic Settings

Wireless Mode	AP	
Wireless Network Mode	Mixed	
Wireless Network Name (SSID)	dd-wrt	
Wireless Channel	6 - 2.437 GHz	
Wireless SSID Broadcast	● Enable ○ Disable	
Sensitivity Range (ACK Timing)	2000	(Default: 2000 meters)

Now enable the Chillispot service and configure it as it is shown on the picture below.



Chillispot

Chillispot	● Enable ○ Disable
Separate Wifi from the LAN Bridge	● Enable ○ Disable
Primary Radius Server IP/DNS	192.168.0.3
Backup Radius Server IP/DNS	192.168.0.3
DNS IP	192.168.0.1
Remote Network	192.168.182.0/24
Redirect URL	https://192.168.0.3/hotspotl
Shared Key	testing123
DHCP Interface	WLAN
Radius NAS ID	dd-wrt
UAM Secret	secret
UAM Any DNS	0
UAM Allowed	
MACauth	○ Enable ● Disable
Additional Chillispot Options	coaport 3779 radiuslisten 192.168.0.50

- **Chillispot** – Activate the Chillispot service.
- **Separate Wifi from the LAN bridge** – Enable the Hotspot server on the WLAN interface.
- **Primary and secondary RADIUS servers** – Enter the Radius Manager server IP in both fields.
- **DNS IP** – A valid DNS server address.
- **Remote network** – Defines the Hotspot client network. Set it to 192.168.182.0/24.
- **Redirect URL** – Defines the Hotspot login page. DD-WRT has no own login page, a remote HTTP server is required. Begin this line with **https://** or **http://**. In our example the complete URL is https://192.168.0.3/hotspotlogin.php. You can find a working *hotspotlogin.php* file in Radius Manager installation archive. Install it on your WEB server.
- **Shared key** – The shared RADIUS secret key, as defined in Radius Manager NAS setup form.
- **DHCP interface** – Select the interface to connect the Hotspot clients. We want to set up a Wireless Hotspot server, so select **WLAN**. You can also select LAN & WLAN here if You want to

connect the clients with Ethernet cable. WAN interface cannot be selected; it is used to connect the router to the Internet.

- **RADIUS NAS ID** – Define it freely to identify your DD-WRT router in RADIUS requests.
- **UAM secret** – This entry should match the secret key defined in *hotspotlogin.php or hotspotlogin.cgi*. The default is "**secret**".
- **UAM any NAS** – Leave it blank.
- **UAM allowed** – Leave it blank.
- **MAC auth**. – Disabled. Currently unsupported.
- **Additional Chillispot options** – Define the **coaport** and **radiuslisten** directives here.

Coaport is required to accept POD packets (remote disconnection), while **radiuslisten** is necessary to send the correct NAS IP address in RADIUS requests. Set **radiuslisten** to NAS IP address (in this example it is 192.168.0.50 – the real address of the DD-WRT device).

After saving and activating the configuration, DD-WRT will generate the Chillispot configuration file and tries to start the Chilli service. If the Hotspot server is not starting You can debug it in Telnet or SSH session. Check the Chilli service PID and the configuration file. If the configuration entries are invalid, Chilli service will not start but no error is reported by the WEB GUI.

You can see the following message in Telnet session if Chilli service is running properly:

```
~ # ps | grep chilli
 4124 root       4840 S   /usr/sbin/chilli -c /tmp/chilli.conf
```

The generated configuration file is located in */tmp* folder.

Notes

Chillispot doesn't support IP address based remote disconnection request (POD), only user names are supported. If You have more than one online session of a specific user, You cannot disconnect all sessions. Always set **simultaneous-use = 1** for every Chillispot account in ACP / Edit user form if You need the remote disconnection function.

Cisco

Radius Manager supports the following features on a Cisco NAS:

1. **RADIUS PPP authentication**, **authorization** and **accounting** (PPPoE, PPPtP, L2tP).
2. User **data rate** management.
3. Automatic **disconnection** of expired accounts.
4. Definable **simultaneous connection** count.
5. PPP **static IP** address.

An IOS version with **AAA new model** and **PPPoE / PPtP** support is required (**vpdn-group** or **bba-group**). In this chapter we'll describe the RADIUS specific Cisco configuration entries.

Enter the following directives to enable the AAA function on your Cisco NAS:

```
aaa new-model
aaa authentication ppp default group radius
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 configuration above controls the AAA features on Cisco. You have to set up the proper **IP pools** with local or public addresses, enable **NAT**ing of local addresses etc. In the example above we use DNS server address 192.168.0.3 and RADIUS server address 192.168.0.3. Substitute these values with your own data. Also select the correct Ethernet interface names.

If You need a **PPPoE service**, set up the correct interface to listen to PPPoE calls (**pppoe enable**).

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

The following data rate limitation modes are supported:

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

Use the following commands to display the current data rates 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**, define **policy-map**s in Cisco (upload, download). Also enter the same policy-maps in ACP /

Edit service. A valid 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
```

For more information please consult the Cisco website on www.cisco.com.

StarOS

Radius Manager supports the following StarOS v2 / v3 services:

- Full **PPPoE** support
- Limited **acccess list** support

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

We recommend to use Star v2 server edition. In StarOS You cannot enable more than one simultaneous connection for any user. StarOS PPPoE system doesn’t support remote disconnection based on IP address. In StarUtil the only supported reference is the username. Always set simultaneous-use = 1 for all StarOS clients (ACP / Edit users form).

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

1. Set the specific **interface** to listen PPPoE request
2. Enable and **configure PPPoE service**
3. **Acivate PPPoE** service
4. Enable **RADIUS** authentication
5. Configure **firewall**
6. **Save** and **activate settings**

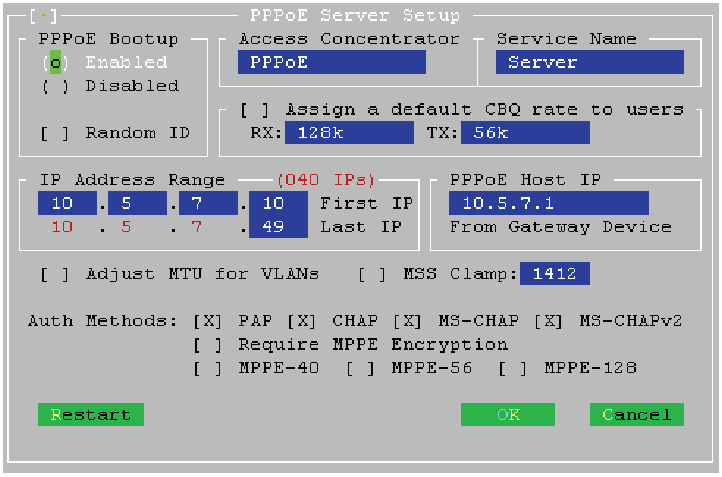
PPPoE server

1. Select **interfaces / [interface name] / listen to pppoe requests: yes** to configure a specific interface as PPPoE server.

2. PPPoE server configuration dialog can be invoked with the menu option

services / pppoe server / bootup/configuration settings

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.

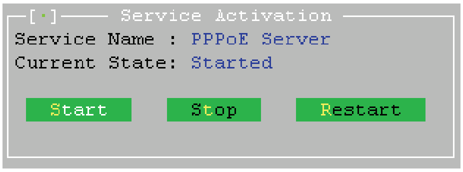


```
[·]                    PPPoE Server Setup
 PPPoE Bootup     Access Concentrator    Service Name
 (o) Enabled      PPPoE                  Server
 ( ) Disabled
                  [ ] Assign a default CBQ rate to users
 [ ] Random ID     RX: 128k        TX: 56k

 IP Address Range   (040 IPs)       PPPoE Host IP
 10 . 5 . 7 . 10   First IP         10.5.7.1
 10 . 5 . 7 . 49   Last IP          From Gateway Device

 [ ] Adjust MTU for VLANs    [ ] MSS Clamp: 1412

Auth Methods: [X] PAP [X] CHAP [X] MS-CHAP [X] MS-CHAPv2
              [ ] Require MPPE Encryption
              [ ] MPPE-40  [ ] MPPE-56  [ ] MPPE-128

 Restart                                  OK      Cancel
```

Select the compatible authentication methods for your CPE devices. PAP is unencrypted. The recommended authentication methods are **CHAP**, **MS-CHAP** and **MS-CHAP v2**. As fallback PAP also can be enabled.

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

services / pppoe server / service activation



```
Service Name : PPPoE Server
Current State: Started
```

4. Enable RADIUS authentication with menu option

services / pppoe server / radius authentication setup

Define the following parameters (assuming your RADIUS server's IP address is 192.168.0.3 and using the 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 mandatory. You can optionally set the retry count, timeout etc.

5. You have to **masquerade** the PPPoE pool if it consists of local address. Invoke the NAT editor with option

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

6. Add 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 on the WAN interface **ether1**. Always select the correct WAN interface.

Save the settings and activate the changes.

7. Select **file / activate changes** to save your settings and activate PPPoE service. Also activate the script changes with option

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

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

RADIUS access list

Radius Manager has limited StarOS RADIUS access list compatibility.

Unfortunately, when a wireless client gets connected using RADIUS access list, StarOS doesn’t send only the access request, but it also sends the accounting information. 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 to this when using the feature.

Use the access list editor to enable the access list support on a specific interface. Invoke it with the option

wireless / [interface name] / access control list editor

Define the default action for handling the 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 connect to SSID.

Notes on StarOS compatibility

- Radius Manager is **fully compatible** with StarOS PPPoE server.
- Radius Manager has **limited compatibility** with StarOS RADIUS Access List system.
- Radius Manager is **not compatible** with StarOS Hotspot system. StarOS sends incorrect NAS IP address in RADIUS requests, doesn’t accept remote disconnect message (POD), sends accounting information in wrong format (upload and download are exchanged) and doesn’t update the accounting data in regular intervals.

If You need a fully functional and free Hotspot system, install Chillispot 1.1.0 on your Linux server. It supports all features which are missing from the StarOS Hotspot system.

PfSense

Radius Manager supports a pfSense NAS. pfSense has a built in Chillispot captive portal which is fully controllable with RADIUS.

The following features are supported:

- Authentication
- Accounting
- Data rate setting per individual users
- Download traffic limitation
- Upload traffic limitation
- Combined traffic limitation
- Online time limitation
- Presettable account expiry date

Restrictions:

- pfSense **does not support remote disconnection** with standard POD packets, instead it uses reauthentication technique, which has some drawbacks over the POD system.
- Due to pfSense uses reauthentication to check the validity of the logged accounts, at least **sim-use = 2** has to be set for every pfSense user in Radius Manager. Sim-use = 1 will result immediate disconnection of the user when the first reauthentication packet arrives to RADIUS (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 manual. It covers the Radius Manager specific configuration details only. For more pfSense informations visit the official website on www.pfsense.com

The following steps are necessary to configure the pfSense Hotspot system:

- Configure **interfaces** (WAN and LAN)
- Configure **DNS**
- Configure **DHCP server**
- Configure **captive portal**

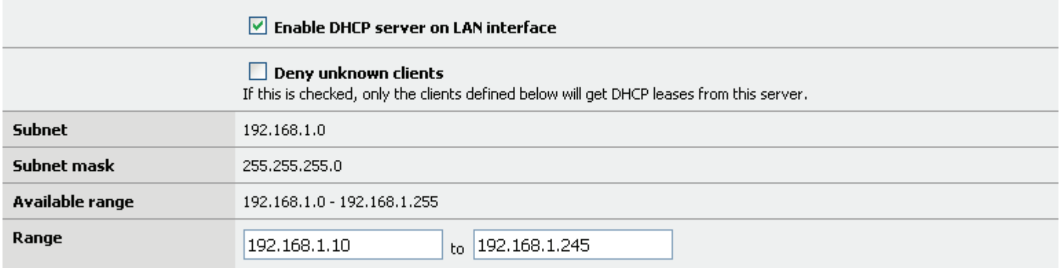
Configuring the network interfaces and DNS

Set the following parameters in the configuration console:

1. **WAN address –** Enter a static WAN address. Radius Manager can't communicate with NAS if dynamic WAN address is used.
2. **LAN address –** It is the gateway of your Hotspot clients. In this example we'll use 192.168.1.1 /24.
3. **Default gateway** – Set the correct gateway to reach the world.
4. **DNS server** – Enter a valid DNS server IP address.

Configuring the DHCP server

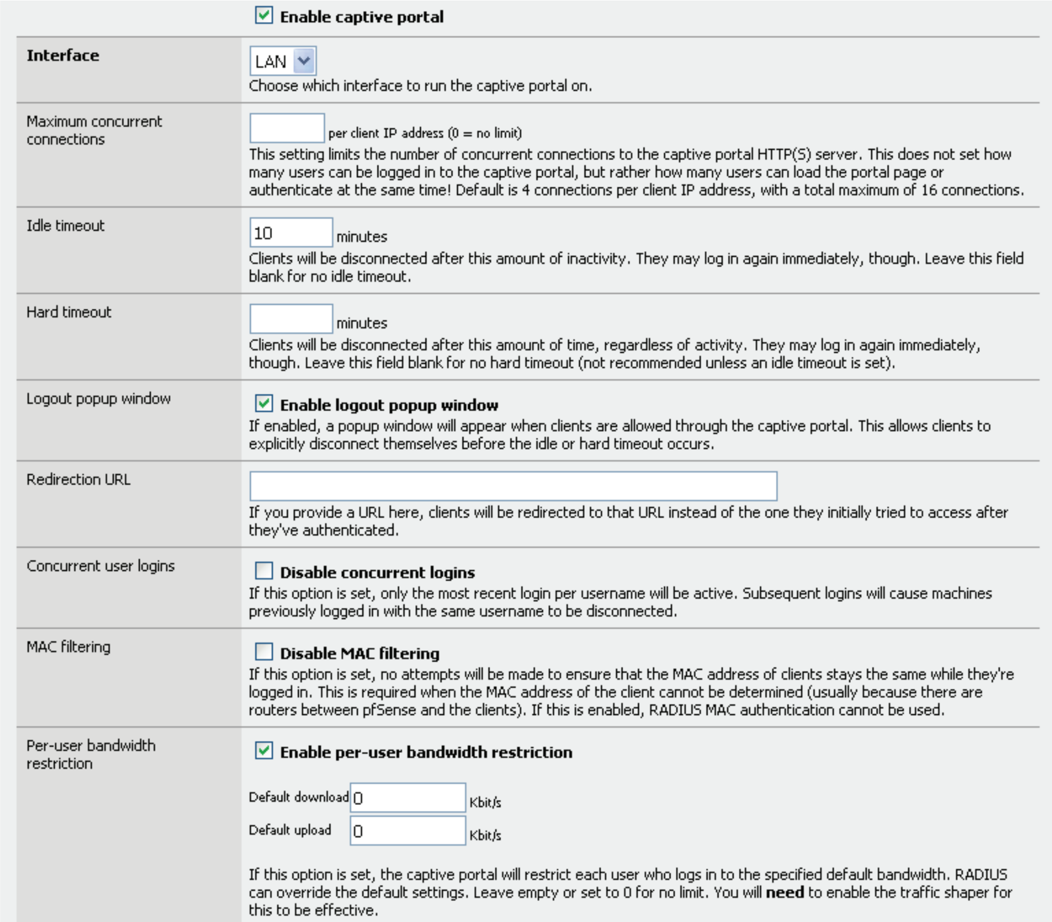
In WEB configurator open the **DHCP configuration** dialog, selecting the *Services / DHCP server* menu option. Enter a valid network range and enable the DHCP server on the LAN interface as it is shown on the picture below. Ensure the LAN IP address is located on the same subnet.



	☑ **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

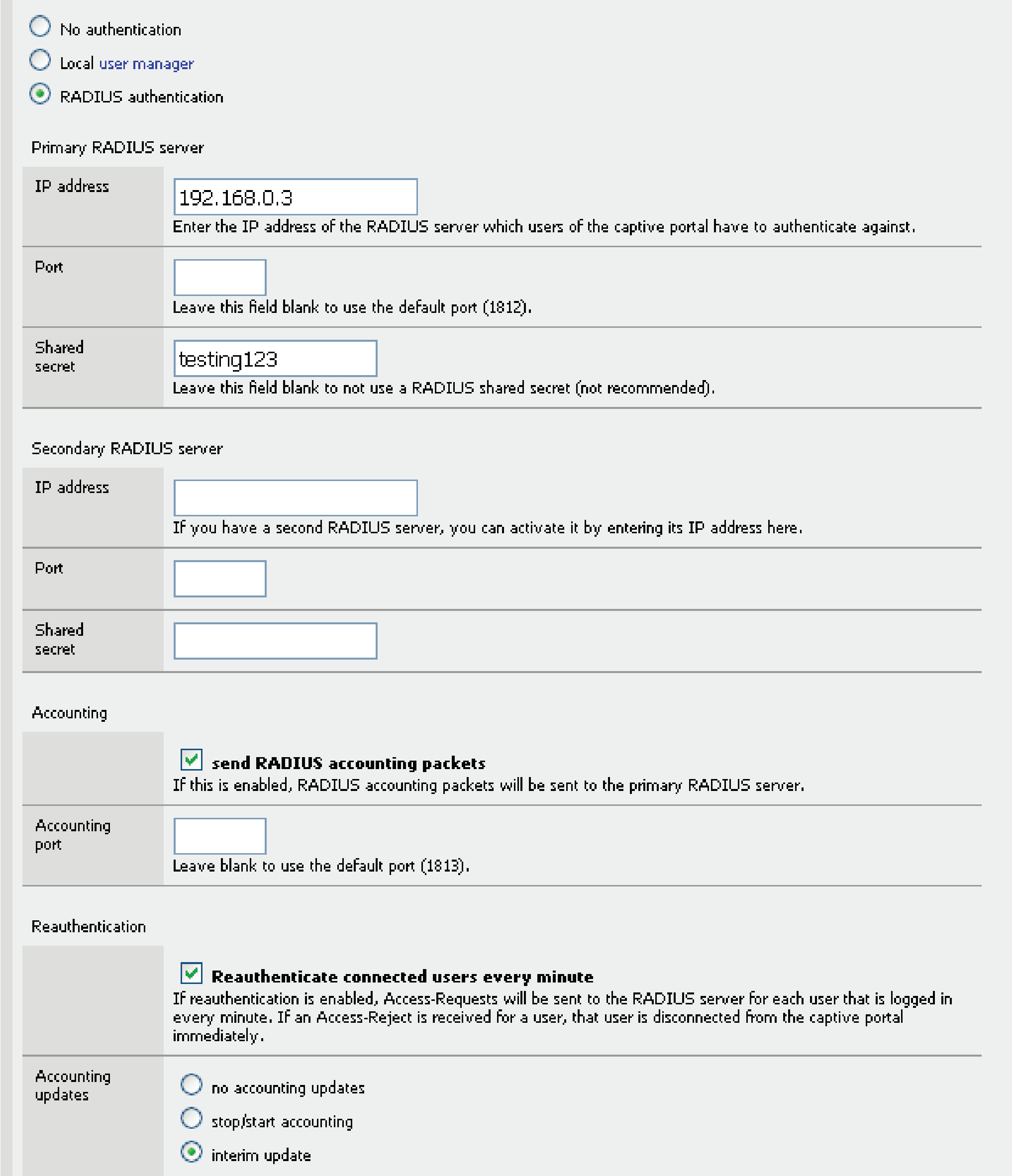
Configuring the captive portal

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



	☑ **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.

1. Open the **Captive portal options** (Services / Captive portal)
2. **Enable** the captive portal with checkbox
3. Select the **interface** to which the Hotspot clients will connect
4. Set **idle timeout** to 10 minutes
5. Enable logout **popup window** with checkbox
6. Enable per-user **bandwidth** restriction
7. Select **RADIUS** authentication
8. Enter the primary **RADIUS server** IP address
9. Enter the **shared secret**
10. Check “**Send RADIUS accounting packets**”
11. Check “**Reauthenticate connected users every minute**”
12. Select accounting updates “**Interim update**”



○ No authentication

○ Local user manager

● RADIUS authentication

Primary RADIUS server

IP address	192.168.0.3 Enter the IP address of the RADIUS server which users of the captive portal have to authenticate against.
Port	Leave this field blank to use the default port (1812).
Shared secret	testing123 Leave this field blank to not use a RADIUS shared secret (not recommended).

Secondary RADIUS server

IP address	If you have a second RADIUS server, you can activate it by entering its IP address here.
Port	
Shared secret	

Accounting

	☑ **send RADIUS accounting packets** If this is enabled, RADIUS accounting packets will be sent to the primary RADIUS server.
Accounting port	Leave blank to use the default port (1813).

Reauthentication

	☑ **Reauthenticate connected users every minute** If reauthentication is enabled, Access-Requests will be sent to the RADIUS server for each user that is logged in every minute. If an Access-Reject is received for a user, that user is disconnected from the captive portal immediately.
Accounting updates	○ no accounting updates ○ stop/start accounting ● interim update

CTS SETUP

Radius Manager has a special feature: the **Connection Tracking System**. It is available in CTS and higher license levels. The CTS system logs all TCP and UDP connections initiated by the registered (online) users.

When You install Radius Manager with CTS module enabled it will use the default CTS database (CONNTRACK). It is strongly recommended to prepare a separate database host for the CONNTRACK database, due to the enormous amount of data stored every day (100-500 MB/day or more). Fast disks are also required to store the data in real time. Radius Manager periodically sends the traffic data to CONNTRACK database (typically in every 5–60 seconds).

You need a Mikrotik router in order to use the CTS feature. It can be:

1. A same router to which the PPP and Hotspot users are connected or
2. A separate router which passes through the traffic.

If You select the second option, You can’t masquerade the clients on PPP / Hotspot server and cannot use transparent proxy. You should ensure that all packets will go through the traffic logger Mikrotik with their original IP addresses. Masquerading can be done after the packets have been processed by the CTS logger.

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

Complete the following steps to enable CTS on a Mikrotik router.

1. Add the following firewall rule to the filter chain:

```
/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 CTS logging on Linux by executing the **rmconntrack** command in debug mode:

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

You have to see how the logging data arrives to Linux when an online user’s UDP or TCP packet is going through the logger Mikrotik.

DOCSIS SETUP

This chapter describes how to configure a Radius Manager **DOCSIS DHCP server**. You can skip this chapter if You have no Radius Manager DOCSIS license available.

The description below covers the CentOS 6 Linux system.

1. First at all install the **tftp server** package:

```
[root@localhost]# yum install tftp-server
```

2. Edit */etc/xinetd.d/tftp,* set **disable = no** and enter the correct **tftp boot file path**:

```
service tftp
{
        socket_type  = dgram
        protocol     = udp
        wait         = yes
        user         = root
        server       = /usr/sbin/in.tftpd
        server_args  = -s /var/www/html/radiusmanager/tftpboot
        disable      = no
        per_source   = 11
        cps          = 100 2
        flags        = IPv4
}
```

Restart xinetd to actualize the changes**:**

```
[root@localhost]# service xinetd restart
```

3. Select the appropriate DHCP server configuration template (**dhcpd.conf-bridge** or **dhcpd.conf-route**) which fits your system configuration (routing or bridge mode CMTS) and **rename** it to **dhcpd.conf**. These files are located in */var/www/html/radiusmanager/config* directory.

4. Set the correct **owner** on *dhcpd.conf*:

```
[root@localhost]# chown apache /var/www/html/radiusmanager/config/dhcpd.conf
```

5. Create a **symbolic link** from *dhcpd.conf* to */etc/dhcpd.conf*:

```
[root@localhost]# ln -s /var/www/html/radiusmanager/config/dhcpd.conf /etc/dhcpd.conf
```

6. **Uninstall** the **DHCP server** package (if already installed):

```
[root@localhost]# rpm -e dhcp
```

7. Install **dhcpd v 3** in */usr/local/sbin* directory. The file is available from:

dmasoftlab.com/cont/downloads

Please note, only **this version will work properly**. Do not try to use different DHCP server versions.

Set **755** permission on **dhcpd** binary file to make it executable:

```
[root@localhost]# chmod 755 /usr/local/sbin/dhcpd
```

8. Install the DHCP **init script** in */etc/init.d* and set the correct permissions. The file is included in Radius Manager installation archive (*rc.d/centos/dhcpd*).

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

Enable DHCP service startup at boot time:

```
[root@localhost]# chkconfig --add dhcpd
```

9. **Start** the DHCP server as service:

```
[root@localhost]# service dhcpd restart
Shutting down dhcpd:                    [FAILED]
Starting dhcpd:                         [  OK  ]
```

It will create the directory for the lease file (*/var/state/dhcp/dhcpd.leases*).

10. **Install** the packages which are required by the **docsis utility**:

```
[root@localhost]# yum install bison net-snmp-devel flex
```

11. Build the **docsis utility**. The sources are available from:

dmasoftlab.com/cont/downloads

```
[root@localhost]# ./configure
[root@localhost]# make
[root@localhost]# make install
```

Test it from shell:

```
[root@localhost]# docsis
DOCSIS Configuration File creator, version 0.9.6
Copyright (c) 1999,2000,2001 Cornel Ciocirlan, ctrl@users.sourceforge.net
Copyright (c) 2002,2003,2004,2005 Evvolve Media SRL, docsis@evvolve.com
```

It should display the usage information.

DHCP server configuration file

The following DOCSIS setups are possible:

- **Routing mode** (Motorola BSR series, Cisco UBR series etc.)
- **Bridge mode** (Arris etc.)

This manual doesn’t cover the configuration steps of CMTS. You can find it in the manual which shipped with your CMTS.

For every CMTS type define the common parameters in **dhcpd.conf** file. It is located in */var/www/html/radiusmanager/config* directory (You can also access it via */etc/dhcpd.conf*).

```
authoritative;
option domain-name “localdomain”;
option domain-name-servers 8.8.8.8;
option time-servers 192.53.103.108;
ddns-update-style none;
min-lease-time 3600;
default-lease-time 3600;
max-lease-time 3600;
log-facility local6;
```

3600 seconds lease time (1 hour) is required to enable automatic disconnection of expired cable modems. Be sure to set the correct **DNS** and **NTP** servers. **DNS** is **essential**, while without NTP server the system can work (but the modems will report warning messages).

Routing mode setup

Complete the following steps to configure a **routing mode** DHCP service. First, define the listening interface:

```
# interface eth0
subnet 192.168.0.0 netmask 255.255.255.0 {
}
```

Define the **CM IP pool**. The CM gateway is the cable interface of the CMTS (10.0.0.1 in this example):

```
# cm
subnet 10.0.0.0 netmask 255.255.0.0 {
    option routers 10.0.0.1;
}
```

Define the **CPE IP pool**. The CPE gateway is the cable interface of the CMTS (10.15.0.1 in this example):

```
# cpe
shared-network cpe {
    subnet 10.15.0.0 netmask 255.255.255.0 {
        option routers 10.15.0.1;
        range dynamic-bootp 10.15.0.2 10.15.0.254;
    }
}
```

Bridge mode setup

The following part explains how to configure a **bridge mode** DHCP server.

First, define a class to differentiate the CM and CPE requests:

```
class “cm” {
#    match if (
#            (binary-to-ascii(16, 8, “:”, substring(hardware, 1, 3)) = “0:13:71”) or
#            (binary-to-ascii(16, 8, “:”, substring(hardware, 1, 3)) = “0:13:72”)
#    );

    match if substring(option vendor-class-identifier,0,6) = “docsis”;

#    log(info, option vendor-class-identifier );
#    log(info, binary-to-ascii(16, 8, “:”, substring(hardware, 1, 6)) );
}
```

In most cases the **vendor-class-identifier** string is enough to set. In special cases (if the system is unable to recognize the CM requests using the **vendor-class-identifier** string) use the MAC address matching mechanism. Uncomment the complete “*match if (...)”* block.

Define the **CM** and **CPE IP pools**:

```
shared-network cm-cpe {
   subnet 192.168.0.0 netmask 255.255.255.0 {
   }

   subnet 10.0.0.0 netmask 255.255.0.0 {
      option routers 10.0.0.1;
   }

   subnet 10.15.0.0 netmask 255.255.255.0 {
      option routers 10.15.0.1;
      pool {
         deny members of “cm”;
         range dynamic-bootp 10.15.0.2 10.15.0.254;
      }
   }
}
```

In this example the listening interface has IP address 192.168.0.x, the CM IP pool is 10.0.0.0/16, the CPE IP pool is 10.15.0.0/16.

The **gateways** (CM and CPE) are configured **on the router**. Don’t forget, in this setup the CMTS is a pure bridge device, it doesn’t do any routing. It has only one IP address (or no one if You configure it via a serial cable).

Testing

Now You can try to run **dhcpd** in debug mode to see the incoming DHCP requests:

```
[root@localhost]# dhcpd -d
Internet Software Consortium DHCP Server V3.0
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Wrote 0 leases to leases file.
Listening on LPF/eth0/00:00:e8:ec:8a:e8/192.168.0.0/24
Sending on   LPF/eth0/00:00:e8:ec:8a:e8/192.168.0.0/24
Sending on   Socket/fallback/fallback-net
```

The command should report no errors. The DHCP server is ready to serve CM and CPE requests. When DHCP server is running in daemon mode, the log messages are sent to **syslog** (*/var/log/messages*).

ADDITIONAL SETUP

Log files

After a certain time FreeRadius log files become enormously big (10-30 MBs). The Linux filesystem can't seek fast enough to the end of the logfile to add new lines, causing degraded system performance and / or RADIUS timeout errors. The logfile has to get stripped regularly to avoid such problems.

Copy *etc/logrotate/radiusd* from radiusmanager tar archive to */etc/logrotate.d* on Linux to enable the automatic logrotation of *radiusd.log*. Radius Manager installer does this job automatically. The included logrotate script is CentOS and Ubuntu compatible. With slight modification it can also be used on other systems.

Starting Radius Manager daemons at boot time

Radius Manager system supports automatic startup for daemons: *radiusd*, *rmpoller* and *rmconntrack*. The installer copies the required scripts to */etc/init.d* directory, sets the required permissions and enables automatic startup of *radiusd*, *rmpoller* and *rmconntrack* daemons.

If You have installed the system in manual mode, copy *rmpoller*, *rmconntrack* and *[ubuntu]/radiusd* or *[centos]/radiusd* files from Radius Manager installation archive to */etc/init.d* directory.

Set **755 permission** on all scripts:

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

The following methods are available to enable automatic service startup:

- Use **Webmin**
- Create **symbolic links** manually
- Use **chkconfig** command (CentOS)
- Use **update-rc.d** command (Ubuntu)

On CentOS issue the following commands:

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

On Ubuntu the commands are:

```
[root@localhost]# update-rc.d rmpoller defaults 99
[root@localhost]# update-rc.d rmconntrack defaults 99
[root@localhost]# update-rc.d radiusd defaults 99
```

Remote UNIX host synchronization

Radius Manager is able to synchronize UNIX accounts on a remote Linux host with RADIUS accounts. Passwordless SSH login is required on the remote host to enable the remote UNIX host synchronization. The following components are required:

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

The following steps are required in order to set up the passwordless SSH login.

1. Generate a OpenSSH 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 the 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:
```

In this example 192.168.0.4 is a **remote server**. The *.ssh* subfolder should be available on the remote host in */root* before issuing the command. Create the *.ssh* folder manually if not present.

After completing this operation You can test the passwordless SSH access to the remote server with the following command:

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

Rootexec permission problem

On some Linux systems (due to the system security) Radius Manager installer is unable to set

4755 permission on *rootexec* binary. Issue the following command to fix it:

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

Fine tuning the Apache WEB server

Edit the Apache configuration to enable the use of **.htaccess** files.

On **CentOS** edit */etc/httpd/conf/httpd.conf* and set **AllowOverride All** (instead of **AllowOverride None**) in *<Directory "/var/www/html">* section*:*

```
<Directory "/var/www/html">
   AllowOverride All
```

On **Ubuntu 10-13** the configuration file is */etc/apache2/sites-enabled/000-default*. Set **AllowOverride All** in *<Directory />* and *<Directory /var/www/>* sections*:*

```
<Directory />
   Options FollowSymLinks
   AllowOverride All
</Directory>
<Directory /var/www/>
   Options Indexes FollowSymLinks MultiViews
   AllowOverride All
   Order allow,deny
   allow from all
</Directory>
```

On **Ubuntu 14** the following snippet should be added to */etc/apache2/sites-available/000-default.conf*, right after **DocumentRoot /var/www/html**:

```
<Directory /var/www/html>
   AllowOverride All
</Directory>
```

Restart Apache to actualize the changes.

REFERENCE

Radius Manager configuration files

system_cfg.php

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

```
// database credentials

define(“db_host”, “localhost”);
define(“db_base”, “radius”);
define(“db_user”, “radius”);
define(“db_psw”, “radius123”);
define(“db_host_cts”, “localhost”);
define(“db_base_cts”, “conntrack”);
define(“db_user_cts”, “conntrack”);
define(“db_psw_cts”, “conn123”);
```

- **db_host** – RADIUS database host name or IP address
- **db_base** – RADIUS database name
- **db_user** – RADIUS database user name
- **db_psw** – RADIUS database password
- **db_host_cts** – CONNTRACK database host name or IP address
- **db_base_cts** – CONNTRACK database name
- **db_user_cts** – CONNTRACK database user name
- **db_psw_cts** – CONNTRACK database password

```
  // system paths and files

define(“radman_dir”, “/var/www/html/radiusmanager”);
define(“raddb_dir”, “/usr/local/etc/raddb”);
define(“tftp_dir”, “tftpboot”);
define(“docsis_keyfile”, “docsis_keyfile”);
define(“docsis_template”, “docsis_template”);
define(“clients_conf”, “clients.conf”);
define(“dhcpd_conf”, “dhcpd.conf”);
define(“leases_file”, “/var/state/dhcp/dhcpd.leases”);
define(“lang_dir”, “lang”);
define(‘config_dir’, ‘config’);
define(“invoice_dir”, “invoice”);
define(‘tmp_images’, ‘tmpimages’);
define(“baseurl”, “http://192.168.0.3/radiusmanager”);
```

- **radman_dir** – Full path of Radius Manager WEB content
- **raddb** – *raddb* directory full path
- **tftp_dir** – TFTP boot files relative path
- **docsis_keyfile** – DOCSIS keyfile name
- **docsis_template** – DOCSIS TFTP template name
- **clients_conf** – Name of *clients.conf* file

- **dhcpd_conf** – DHCP configuration file name
- **leases_file** – DHCP leases file full path
- **lang_dir** – Relative path for language files relative path
- **config_dir** – Folder for configuration files
- **invoice_dir** – Invoice template relative path
- **tmp_images** – Temporary images relative path
- **baseurl** – Complete URL of Radius Manager

```
// system definitions

define(“admin_user”, “admin”);
define(‘def_syslang’, ‘English’);
define(“rootexec_psw”, “12345”);
define(‘httpd_user’, ‘apache’);
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(“cardsernum_integers”, 12);
define(“cardseries_padding“, 4);
define(“card_pin_len”, 8);
define(“card_psw_len”, 4);
define(“ias_pin_length”, 8);
define(“ias_psw_length”, 4);
define(“rndchars”, “0123456789ABCDEFGHIJKLMNOPQRSTVWXYZ”);
define(‘rndcardpin’, ‘0123456789’);
define(‘rndcardpass’, ‘0123456789’);
define(“rndstring_len”, 4);
define(“max_smsnums”, 3);
define(“max_pinfails”, 3);
define(“max_verifyfails”, 3);
define(‘max_sameselfreg’, 3);
define(“quickjump_max_pages”, 10);
define(“rows_per_page”, 50);
define(“csv_max_rows”, 1000000);
define(“cc_years”, 5);
define(“session_timeout”, 15);
define(“regexp_username”, ‘/^[a-z0-9._]+$/’);
define(“regexp_managername”, ‘/^[a-z0-9._]+$/’);
define(‘regexp_email’, ‘/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,4})$/’);
define(“regexp_mac”, ‘/^[:a-z0-9._]+$/’);
define(“regexp_psw”, ‘/^[a-zA-Z0-9._]+$/’);
define(“keep_connlog”, 190);
define(“keep_syslog”, 30);
define(“keep_actsrv”, 1);
define(“ping_timeout”, 1);
define(“pswact_len_email”, 60);
define(“pswact_len_sms”, 8);
define(“newpsw_len”, 4);
define(“grp_dec_inv”, true);
define(“default_simuse”, 1);
define(“cmperthread”, 50);
define(“cm_community”, “private”);
define(“mt_login_delay”, 200000);
define(‘colsel_itemperrow’, 4);
```

- **admin_user** – Name of Radius Manager super user
- **def_syslang** – Default system language (fallback)
- **rootexec_psw** – Password for rootexec program.
- **httpd_user** – Apache user name
- **nas_port_mt** – Radius incoming port for Mikrotik. It is global for all Mikrotik NASs
- **nas_port_chilli** – Radius incoming port for Chillispot. It is global for all Chillispot NASs
- **nas_port_cisco** – Radius incoming port for Cisco. It is global for all Cisco NASs
- **hotspot_ip** – IP or URL of Hotspot captive portal
- **no_limit_date** – Date for unlimited Unix account expiration (should be in future)
- **max_card_quantity** – The maximum number of cards which can be generated at once
- **cardsernum_integers** – Card serial number length in CSV files
- **cardseries_padding** – Number of digits in card series
- **card_pin_len –** PIN code length of prepaid cards
- **card_psw_len –** Password length of prepaid cards
- **ias_pin_length –** IAS user name length
- **ias_psw_length –** IAS password length
- **rndchars** – Default random characters
- **rndcardpin** – Random characters in card PIN codes
- **rndcardpass** – Random characters in card passwords
- **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
- **max_sameselfreg** – Maximal number of same self registered account names
- **quickjump_max_pages** – Number of pages in quickjump links
- **rows_per_page** – Number screen rows per page
- **csv_max_rows** – Number of rows in CSV file
- **cc_years** – How many years to display in CC expiration listboxes
- **session_timeout** – PHP session timeout in minutes
- **regexp_username** – Regular expression for user name validation
- **regexp_managername** – Regular expression manager name validation
- **regexp_email** – Regular expression for email address validation
- **regexp_mac** – Regular expression MAC address validation
- **regexp_psw** – Regular expression for password validation
- **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 days to keep the actual service data
- **keep_postauth** – How many days to keep the postauth log data
- **ping_timeout** – Ping timeout value in seconds
- **pswact_len_email** – Length of new password activation code sent in email
- **pswact_len_sms** – Length of new password activation code sent in sms
- **newpsw_len** – Length of generated password in password recovery
- **grp_dec_inv** – Enable grouping of decimals on invoice forms
- **default_simuse** – Default sim-use value for new users

- **cmperthread** – Number of CMs per thread in cmtspoller module
- **cm_community** – CM community string
- **mt_login_delay** – Delay between Mikrotik API login attempt and response (in microseconds)
- **colsel_itemperrow** – Number of items per row in column selector

```
// SMTP definitions

  define(‘smtp_relay’, ‘localhost’);
  define(‘smtp_port’, 25);
  define(‘smtp_auth’, FALSE);
  define(‘smtp_user’, ‘username’);
  define(‘smtp_psw’, ‘password’);
  define(‘smtp_secure’, ‘’);
  define(‘smtp_charset’, ‘UTF-8’);
  define(‘smtp_debug’, FALSE);
  define(‘smtp_html’, FALSE);
  define(‘mail_from’, ‘admin@myisp.com’);
  define(‘mail_fromname’, ‘Administrator’);
  define(‘mail_newuser’, ‘admin@localhost’);
  define(‘mail_localdomain’, ‘localhost.localdomain’);
```

- **smtp_relay** – SMTP relay host
- **smtp_port** – SMTP port
- **smtp_auth** – Enable SMTP authentication
- **smtp_user** – SMTP user name
- **smtp_psw** – SMTP password
- **smtp_secure** – Secure protocol (TLS, SSL or blank; Gmail requires TLS or SSL)
- **smtp_charset** – Character enconding scheme
- **smtp_debug** – Enable debugging (TRUE or FALSE)
- **smtp_html** – HTML mode (TRUE or FALSE)
- **mail_from** – Sender address
- **mail_fromname** – Sender name
- **mail_newuser** – Self registration notification address
- **mail_localdomain** – Default domain name

```
// 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(‘min_pswhsmac_len’, 4);
define(‘max_pswhsmac_len’, 32);
define(“mobile_minlen”, 6);
define(“mobile_maxlen”, 16);
define(“comment_maxlen”, 30);
```

- **min_username_len** – Minimal user name length

- **max_username_len** – Maximal user name length
- **mac_username_len_mikrotik** – Mikrotik MAC user name length
- **mac_username_len_staros** – StarOS MAC user name length
- **min_psw_len** – Minimal password length
- **max_psw_len** – Maximal password length
- **min_pswhsmac_len** – Minimal Hotspot MAC password length
- **max_pswhsmac_len** – Maximal Hotspot MAC password length
- **mobile_minlen** – Minimal mobile number length (verification)
- **mobile_maxlen** – Maximal mobile number length (verification)
- **comment_maxlen** – Number of haracters in comment field

```
// card PDF export

define(“cards_per_page”, 10);
define(“username_x_pos”, 45);
define(“username_y_pos”, 36);
define(“pdfprint_expiration”, true);
define(“pdfprint_price”, true);
define(“pdfprint_serial”, true);
define(“pdfprint_series”, true);
define(“pdfprint_descr”, true);
define(“psw_x_pos”, 45);
define(“psw_y_pos”, 44);
define(“pin_x_pos”, 33);
define(“pin_y_pos”, 40);
define(“price_x_pos”, 75);
define(“price_y_pos”, 19);
define(“date_x_pos”, 53);
define(“date_y_pos”, 53);
define(“serial_x_pos”, 27);
define(“serial_y_pos”, 61);
define(“series_x_pos”, 54);
define(“series_y_pos”, 61);
define(“descr_x_pos”, 15);
define(“descr_y_pos”, 26);
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”, 10);
define(“price_font_color”, “FFF7A1”);
define(“serial_font_type”, “Times”);
define(“serial_font_size”, 8);
define(“serial_font_color”, “CEDDFF”);
define(“series_font_type”, “Times”);
define(“series_font_size”, 8);
define(“series_font_color”, “CEDDFF”);
define(“srvname_font_type”, “Arial”);
define(“srvname_font_size”, 12);
define(“srvname_font_color”, “DFEFF3”);
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);
```

- **cards_per_page** – Number of cards per A4 sheet
- **username_x_pos** – Horizontal position of user name on classic prepaid cards
- **username_y_pos** – Vertical position of user name on classic prepaid cards
- **pdfprint_expiration** – Enable printing the expiry date
- **pdfprint_price** – Enable printing the price
- **pdfprint_serial** – Enable printing the card serial number
- **pdfprint_series** – Enable printing the card series number
- **pdfprint_descr** – Enable printing the service description
- **psw_x_pos** – Horizontal position of password on classic prepaid cards
- **psw_y_pos** – Vertical position of password on classic prepaid cards
- **pin_x_pos** – Horizontal position of PIN code on refill cards
- **pin_y_pos** – Vertical position of PIN code on refill cards
- **price_x_pos** – Horizontal position of price on cards
- **price_y_pos** – Vertical position of price on cards
- **date_x_pos** – Horizontal position of valid till field on cards
- **date_y_pos** – Vertical position of valid till field on cards
- **serial_x_pos** – Horizontal position of service name on cards
- **serial_y_pos** – Vertical position of service name on cards
- **series_x_pos** – Horizontal position of series on cards
- **series_y_pos** – Vertical position of series on cards
- **descr_x_pos** – Horizontal position of description x on cards
- **descr_y_pos** – Vertical position of description x on cards
- **user_font_type** – PIN and password font typeface
- **user_font_size** – PIN and password font size
- **user_font_color** – PIN and 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
- **serial_font_type** – Serial font typeface
- **serial_font_size** – Serial font size
- **serial_font_color** – Serial font color
- **series_font_type** – Series font typeface
- **series_font_size** – Series font size
- **series_font_color** – Series font color
- **srvname_font_type** – Serial font typeface
- **srvname_font_size** – Serial font size
- **srvname_font_color** – Serial font color
- **card_left_margin** – Left margin
- **card_top_margin** – Top margin

- **card_classic_bg_filename** – Classic prepaid card background image file
- **card_refill_bg_filename** – Refill card background image file
- **card_bg_width** – Prepaid card background image width
- **card_bg_height** – Prepaid card background image height

```
// unix executables

define(“cmd_rootexec”, “/usr/local/sbin/rootexec”);
define(‘cmd_rmlic’, ‘/usr/local/sbin/rmlic’);
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”);
define(“cmd_ping”, “/bin/ping”);
define(“cmd_docsis”, “/usr/local/bin/docsis”);
```

- **cmd_rootexec** – rootexec command with full path
- **cmd_rmlic** – rmlic command with full path
- **cmd_radclient** – Radclient utility with full path
- **cmd_starutil** – Starutil utility command with full path
- **cmd_useradd** – Useradd command with full path
- **cmd_userdel** – Userdel command with full path
- **cmd_chmod** – Chmod command with full path
- **cmd_usermod** – Usermod command with full path
- **cmd_passwd** – Passwd command with full path
- **cmd_edquota** – Edquota command with full path
- **cmd_ping** – Ping command with full path
- **cmd_docsis** – Docsis utility with full path

```
// gradient bars

define(‘GDBAR_WIDTH’, 50);
define(‘GDBAR_HEIGHT’, 3);
define(‘GDBAR_BGCOLOR’, ‘#000000’);
define(‘GDBAR_RED’, ‘#FF0000’);
define(‘GDBAR_YELLOW’, ‘#FFFC00’);
define(‘GDBAR_GREEN’, ‘#00FF00’);
```

- **GDBAR_WIDTH** – Gradient bar width
- **GDBAR_HEIGHT** – Gradient bar height
- **GDBAR_BGCOLOR** – Gradient bar background color
- **GDBAR_RED** – Gradient bar red color
- **GDBAR_YELLOW** – Gradient bar yellow color
- **GDBAR_GREEN** – Gradient bar green color

```
// CM specific

define(‘CM_SCALE_MIN’, 0);
define(‘CM_SCALE_MAX’, 140);
define(‘CM_TXSIGNAL_MIN’, 95);
define(‘CM_TXSIGNAL_MAX’, 115);
define(‘CM_RXSIGNAL_MIN’, 50);
define(‘CM_RXSIGNAL_MAX’, 75);
define(‘CM_SNRDS_MIN’, 0);
define(‘CM_SNRDS_MAX’, 50);
define(‘CM_SNRUS_MIN’, 0);
define(‘CM_SNRUS_MAX’, 35);
```

- **CM_SCALE_MIN** – CM scale start
- **CM_SCALE_MAX** – CM scale end
- **CM_TXSIGNAL_MIN** – CM TX minimal usable signal level
- **CM_TXSIGNAL_MAX** – CM TX maximal usable signal level
- **CM_RXSIGNAL_MIN** – CM RX minimal usable signal level
- **CM_RXSIGNAL_MAX** – CM RX maximal usable signal level
- **CM_SNRDS_MIN** – CM SNR DS minimal level
- **CM_SNRDS_MAX** – CM SNR DS maximal level
- **CM_SNRUS_MIN** – CM SNR US minimal level
- **CM_SNRUS_MAX** – CM SNR US maximal level

```
// WLAN specific

define(‘WLAN_SIGNAL_MIN’, -90);
define(‘WLAN_SIGNAL_MAX’, -65);
define(‘WLAN_SNR_MIN’, 0);
define(‘WLAN_SNR_MAX’, 40);
```

- **WLAN_SIGNAL_MIN** – WLAN minimal signal level
- **WLAN_SIGNAL_MAX** – WLAN maximal signal level
- **WLAN_SNR_MIN** – WLAN minimal SNR
- **WLAN_SNR_MAX** – WLAN maximal SNR

```
// captcha

define(‘CAPTCHA_FONT’, ‘monofont.ttf’);
define(‘CAPTCHA_WIDTH’, 120);
define(‘CAPTCHA_HEIGHT’, 40);
define(‘CAPTCHA_LEN’, 4);
```

- **CAPTCHA_FONT** – Font typface
- **CAPTCHA_WIDTH** – Image width
- **CAPTCHA_HEIGHT** – Image height
- **CAPTCHA_LEN** – Number of characters

```
  // SNMP

define(‘SNMP_CMTS_MAC’,     ‘.1.3.6.1.2.1.10.127.1.3.3.1.2’);
define(‘SNMP_CMTS_IP’,      ‘.1.3.6.1.2.1.10.127.1.3.3.1.3’);
define(‘SNMP_CMTS_STATUS’,  ‘.1.3.6.1.2.1.10.127.1.3.3.1.9’);
define(‘SNMP_CMTS_IFIDX’,   ‘.1.3.6.1.2.1.10.127.1.3.3.1.5’);
define(‘SNMP_CMTS_IFDESCR’, ‘.1.3.6.1.2.1.2.2.1.2’);
define(‘SNMP_CMTS_SNRUS’,   ‘.1.3.6.1.2.1.10.127.1.1.4.1.5’);
define(‘SNMP_CM_RESTART’, ‘.1.3.6.1.2.1.69.1.1.3.0’);
define(‘SNMP_CM_SNRDS’,   ‘.1.3.6.1.2.1.10.127.1.1.4.1.5’);
define(‘SNMP_CM_RXPWR’,   ‘.1.3.6.1.2.1.10.127.1.1.1.1.6’);
define(‘SNMP_CM_TXPWR’,   ‘.1.3.6.1.2.1.10.127.1.2.2.1.3’);
define(‘SNMP_CM_CPEMAC’,  ‘.1.3.6.1.2.1.17.4.3.1.1’);
define(‘SNMP_CM_CPETYPE’, ‘.1.3.6.1.2.1.17.4.3.1.3’);
define(‘SNMP_CM_UPTIME’,  ‘.1.3.6.1.2.1.1.3’);
define(‘SNMP_WLAN_SIGNAL’,  ‘.1.3.6.1.4.1.14988.1.1.1.2.1.3’);
```

- **SNMP_CMTS_MAC** – CM MAC address
- **SNMP_CMTS_IP** – CM IP address
- **SNMP_CMTS_STATUS** – CM status
- **SNMP_CMTS_IFIDX** – interface index
- **SNMP_CMTS_IFDESCR** – interface description
- **SNMP_CMTS_SNRUS** – SNR upstream
- **SNMP_CM_RESTART** – restart CM command
- **SNMP_CM_SNRDS** – CM SNR downstream
- **SNMP_CM_RXPWR** – CM RX power
- **SNMP_CM_TXPWR** – CM TX power
- **SNMP_CM_CPEMAC** – CM CPE MAC address
- **SNMP_CM_CPETYPE** – CM CPE MAC address type
- **SNMP_CM_UPTIME** – CM uptime
- **SNMP_WLAN_SIGNAL** – WLAN signal level

paypal_cfg.php

Radius Manager supports **PayPal Express Checkout**, **PayPal Website Payments Pro** and **PayPal Website Payments Standard** API (www.paypal.com).

- **PayPal Express Checkout** works with premier and business accounts and can be used to PayPal accept balance and CC payments.
- **PayPal Website Payments Pro** requires Pro or better account and works with US / UK merchants only. It supports CC payments only.
- **PayPal Website Payments Standard** can be used for balance and CC payments and it supports multiple merchant countries.

The recommended APIs are PayPal Express Checkout and PayPal Website Payments Pro. We discourage You to use PayPal Website Payments Standard.

PayPal subsystem configures in *paypal_cfg.php* file which is located in the *config* directory. The most important configuration entries 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 parameters:

- **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 configure your PayPal **Sandbox** account. Register a test account, enter the Sandbox credentials in *paypal_cfg.php* and set **TEST_MODE** to **TRUE**. Logging to PayPal developer account is required (in another browser window) when testing the system in Sandbox environment.

An SSL certificate is required to enable the **PayPal Website Payments Standard** API. The next part explains the steps required to generate a such certificate.

Generating Your Private Key Using OpenSSL

Enter the following command to generate your private key. This command generates a 1024-bit RSA private key (*ewp-key.pem*):

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

Generating Your Public Certificate Using OpenSSL

The public certificate requires PEM format. Enter the following command to generate your publicc certificate (*ewp-cert.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]:New York city
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
```

Uploading your public certificate to your PayPal account

1. Log into your **PayPal Business** or **Premier** account
2. Click the **Profile** subtab.
3. In the **Selling Preferences** column, click the **Encrypted Payment Settings** link. The Website Payment Certificates page will appear.
4. Scroll down the page to **Your Public Certificates** section, and click the Add button.
5. The **Add Certificate** page appears.
6. Click the **Browse** button and select the public certificate You want to upload from your local computer (*certs/ewp-cert.pem*).
7. Click the **Add** button.
8. Once the public certificate has been uploaded, it will appear in the **Your Public Certificates** section of the **Website Payment Certificates** page.
9. Copy the associated certificate ID to **DEFAULT_EWP_CERT_ID** field in *paypal_cfg.php*.

Downloading the PayPal public certificate from the PayPal website

1. Log into your **Business** or **Premier** PayPal account.
2. Click the **Profile** subtab.
3. In the **Selling Preferences** column click the **Encrypted Payment Settings** link.
4. Scroll down the page to **PayPal Public Certificate** section.
5. Click the **Download** button and save the file in a secure location on your local computer (*certs/paypal-cert.pem*).

sagepay_cfg.php

Radius Manager system supports SagePay South Africa (www.sagepay.co.za, former NetCash) credit card payment gateway. You need a SagePay merchant account to use this feature.

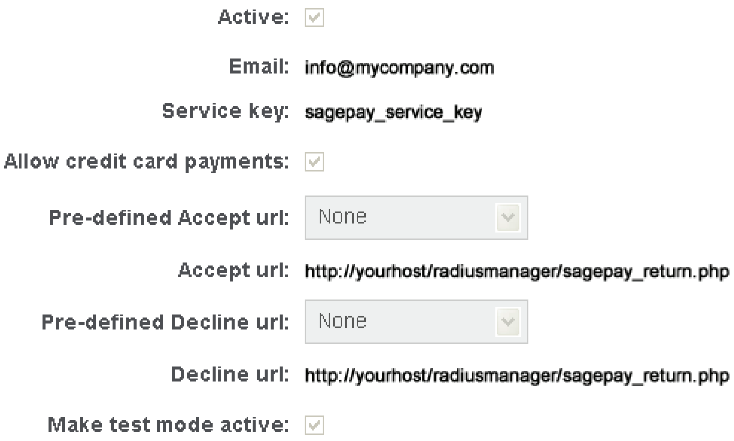
The SagePay module configures in *sagepay_cfg.php* which is located in *radiusmanager/config* directory. The available configuration entries are:

```
// SagePay credentials

define(‘SAGEPAY_SRVKEY’, ‘service_key’);
define(‘SAGEPAY_EMAIL’,  ‘info@mycompany.com’);
```

Description of parameters:

- **SAGEPAY_SRVKEY** – SagePay service key.
- **SAGEPAY_EMAIL** – Email address to receive transaction reports sent bySagePay.

You have to enter the correct **Accept** and **Reject URL**s in SagePay control panel, in Account profile / PayNow section, as shown on the picture below.



Active: ☑

Email: info@mycompany.com

Service key: sagepay_service_key

Allow credit card payments: ☑

Pre-defined Accept url: None

Accept url: http://yourhost/radiusmanager/sagepay_return.php

Pre-defined Decline url: None

Decline url: http://yourhost/radiusmanager/sagepay_return.php

Make test mode active: ☑

payfast_cfg.php

This chapter explains the configuration steps for PayFast online payment gateway. PayFast is a hosted payment solution with HTTP redirection and supports South African merchants.

PayFast module configures in *payfast_cfg.php* which is located in *radiusmanager/config* directory. The available configuration entries are:

```
define(‘PAYFAST_MERCHANT_ID’, ‘your_merchant_id’);
define(‘PAYFAST_MERCHANT_KEY’, ‘your_merchant_key’);
define(‘PAYFAST_PDT_KEY’, ‘your_pdt_key’);

// test or live mode

define(‘PAYFAST_TEST_MODE’, TRUE);

// API URL

define(‘PAYFAST_URL_TEST’, ‘sandbox.payfast.co.za’);
define(‘PAYFAST_URL_LIVE’, ‘www.payfast.co.za’);

// PayFast WEB language

define(‘PAYFAST_LANG’, ‘eng’);

// return URL

define(“PAYFAST_RETURN_URL”, “payfast_return.php”);
```

Description of parameters:

- **PAYFAST_MERCHANT_ID** – Merchant id.
- **PAYFAST_MERCHANT_KEY** – Merchant key.
- **PAYFAST_PDT_KEY** – PDT key.
- **PAYFAST_TEST_MODE** – Set TRUE to enable test mode.
- **PAYFAST_URL_TEST** – URL for test order.
- **PAYFAST_URL_LIVE** – URL for live order.
- **PAYFAST_LANG** – PayFast WEB interface language.
- **PAYFAST_RETURN_URL** – Return URL.

authorizenet_cfg.php

Radius Manager utilizes **Authorize.net** to accept credit cards online (www.authorize.net). The system doesn’t store any data on the local host, instead it forwards the CC data to authorize.net (AIM integration method). Ensure You are running the **HTTP** server in **secure mode** (SSL) when You are working with credit cards!

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

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

define(‘AUTHORIZENET_USERNAME’, ‘login_id’);
define(‘AUTHORIZENET_TRANSKEY’, ‘transaction_key’);
define(“AUTHORIZENET_TEST_MODE”, TRUE);

// default URL’s

define(‘AUTHORIZENET_URL_TEST’, ‘https://test.authorize.net/gateway/transact.dll’);
define(‘AUTHORIZENET_URL_LIVE’, ‘https://secure.authorize.net/gateway/transact.dll’);
```

Description of parameters:

- **AUTHORIZENET_USERNAME** – API user name.
- **AUTHORIZENET_TRANSKEY** – API transaction key.
- **AUTHORIZENET_TEST_MODE** – Set it to TRUE if You use your Authorize.net account in test mode or FALSE if You want to use your live account.
- **AUTHORIZENET_URL_TEST** – The test mode gateway URL. Use the default value here.
- **AUTHORIZENET_URL_LIVE** – The live mode gateway URL. Use the default value here.

dps_cfg.php

DPS Express Payment gateway (www.paymentexpress.com) is available in Radius Manager to accept credit cards online. It supports multiple merchant countries. The system doesn’t store any data on the local host; the CC authorization is done by the DPS site (redirection). When a CC has been processed (success or failure) the browser gets directed back to Radius Manager site.

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

```
define(“DPS_URL”, “https://sec2.paymentexpress.com/pxpay/pxaccess.aspx”);
define(“DPS_USERNAME”, “username”);
define(“DPS_KEY”, “key”);

define(“DPS_RETURN_URL”, “dps_return.php”);
define(“DPS_EMAIL”, “info@mycompany.com”);
```

Description of parameters:

- **DPS_URL** – The payment gateway URL. Use the default value here.
- **DPS_USERNAME** – API user name.
- **DPS_KEY** – API transaction key.
- **DPS_RETURN_URL** – The URL called after the transaction.
- **DPS_EMAIL** – The email address of the merchant.
- **currency_dps** – The available currencies as they are defined in DPS specification.

2co_cfg.php

Radius Manager can utilize **2Checkout.com** online payment provider (www.2checkout.com). It supports multiple countries and currencies and very simple to configure.

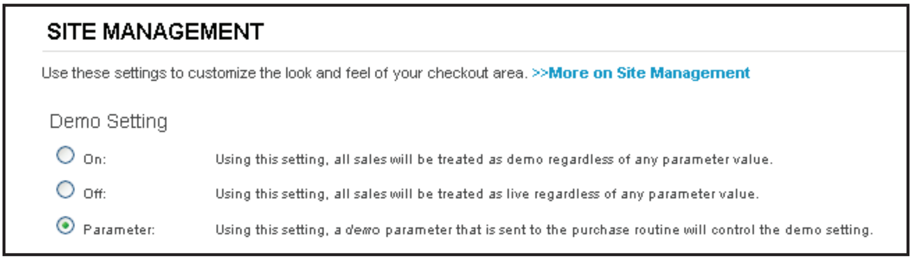
The configuration entries are:

```
// API credentials

define('_2CO_SID', "vendor_id");
define('_2CO_SECRET', "secret_word");

// additional data

define("_2CO_TEST_MODE", TRUE);
define("_2CO_SKIP_LANDING", "1");
```

Description of parameters:

- **_2CO_SID** – Account identifier. Get if from 2Checkout.com.
- **_2CO_SECRET** – Secret transaction key. Get if from 2Checkout.com.
- **_2CO_TEST_MODE** – Enable (TRUE) or disable (FALSE) the test mode. Don't forget to configure the test mode in 2Checkout.com control panel, setting only this variable is not enough.
- **_2CO_SKIP_LANDING** – Do not show the cart review page in transactions.
- **currency_2co** – The available currencies as they are defined in 2Checkout specification.

There are some extra parameters You need to set in your 2CO control panel.



SITE MANAGEMENT

Use these settings to customize the look and feel of your checkout area. >>More on Site Management

Demo Setting

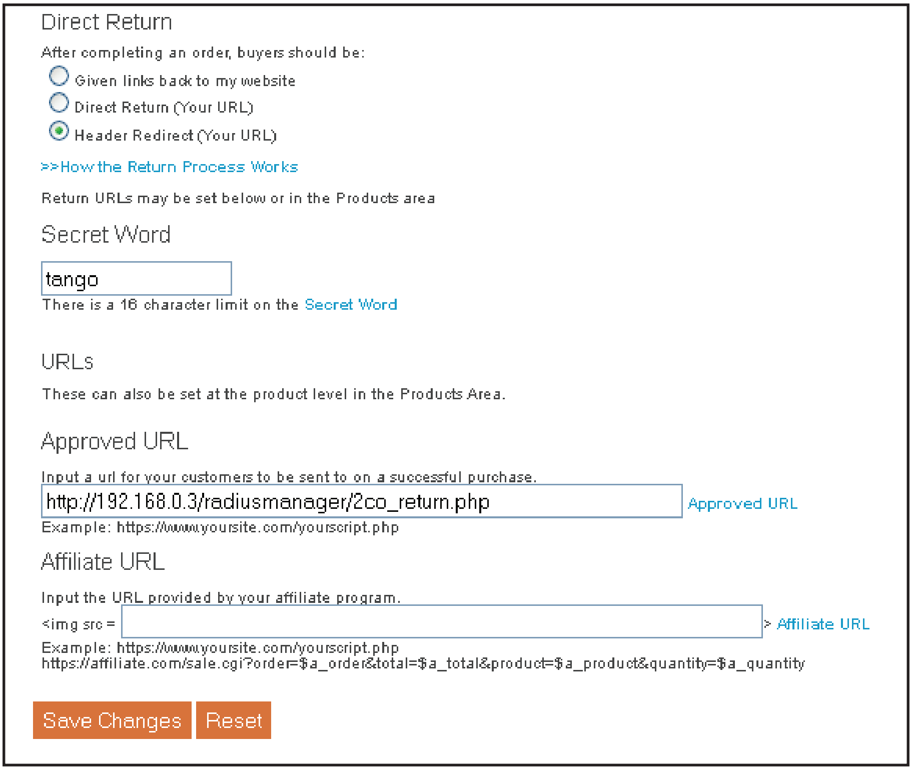
○ On: Using this setting, all sales will be treated as demo regardless of any parameter value.

○ Off: Using this setting, all sales will be treated as live regardless of any parameter value.

● Parameter: Using this setting, a demo parameter that is sent to the purchase routine will control the demo setting.

1. Go to **Account / Site management** and select **Parameter** in **Demo setting.**
2. Scroll down to **Direct return** section and select **Header redirect.**
3. Enter the **secret word** as it is defined in *2co_cfg.php.*
4. In the approved URL field enter the absolute path of your **2co_return.php** file.

Click **Save changes** after completing the form.



Direct Return

After completing an order, buyers should be:

○ Given links back to my website

○ Direct Return (Your URL)

● Header Redirect (Your URL)

>>How the Return Process Works

Return URLs may be set below or in the Products area

Secret Word

tango

There is a 16 character limit on the Secret Word

URLs

These can also be set at the product level in the Products Area.

Approved URL

Input a url for your customers to be sent to on a successful purchase.

http://192.168.0.3/radiusmanager/2co_return.php Approved URL

Example: https://www.yoursite.com/yourscript.php

Affiliate URL

Input the URL provided by your affiliate program.

<img src = > Affiliate URL

Example: https://www.yoursite.com/yourscript.php
https://affiliate.com/sale.cgi?order=$a_order&total=$a_total&product=$a_product&quantity=$a_quantity

citrus_cfg.php

DMA Radius Manager 4.2 supports Citrus Payments (PayUmoney) online payment provider (www.citruspay.com). This payment gateway is available for Indian merchants for accepting payments online. Users can recharge their accounts in UCP with a few simple click.

The configuration entries are:

```
// Citrus merchant credentials

define(‘CITRUS_VANITYURL’, ‘url’);
define(‘CITRUS_SECRET’,    ‘secret’);

// test or live mode

define(‘CITRUS_TEST_MODE’, TRUE);

// default URLs

define(‘CITRUS_URL_TEST’,   ‘https://sandbox.citruspay.com’);
define(‘CITRUS_URL_LIVE’,   ‘https://checkout.citruspay.com/ssl/checkout’);
define(‘CITRUS_RETURN_URL’, ‘citrus_return.php’);
```

Description of parameters:

- **CITRUS_VANITYURL** – It is generated by Citrus payment provider and available in Citrus control panel.
- **CITRUS_SECRET** – It is generated by Citrus payment provider and available in Citrus control panel.
- **CITRUS_TEST_MODE** – Enable (TRUE) or disable (FALSE) the test mode.
- **CITRUS_URL_TEST** – Test mode URL.
- **CITRUS_URL_LIVE** – Live mode URL.
- **CITRUS_RETURN_URL** – The redirection URL after completing the payment.

To configure Citrus payments, register a new account in www.citruspay.com and set the CITRUS_VANITYURL / CITRUS_SECRET variables. For live mode set CITRUS_TEST_MODE to FALSE.

NOTICE

In sandbox test mode CItrus payment gateway cannot complete the online payment. Test mode can be used to confirm the functionality of the payment page in UCP. After entering the card / bank details, Citrus gateway responds with a failure. Enable live mode to complete a real transaction.

radiusmanager.cfg

Radiusmanager.cfg is located in /etc folder. It is the configuration file for Radius Manager **utilities**. The content of *radiusmanager.cfg* is listed below:

```
db_host                   localhost
db_name                   radius
db_user                   radius
db_psw                    radius123
db_host_cts               localhost
db_name_cts               conntrack
db_user_cts               conntrack
db_psw_cts                conn123
db_sock                   /var/lib/mysql/mysql.sock
radman_path               /var/www/html/radiusmanager
def_lang                  English
rootexec_psw              12345
inactivity                10
poller_pause              60
api_pause                 60
cmpoller_pause            300
radclient                 /usr/local/bin/radclient
starutil                  /usr/local/bin/starutil
nas_port_mt               1700
nas_port_chilli           3779
nas_port_cisco            1700
mt_api_port               8728
cts_port                  4950
cts_blocksize             5000
cts_file                  /tmp/rmconnlog
cts_threads               8
cts_flush                 30
cts_username_len          32
cts_allindex              yes
cts_logallip              no
socket_rmconntrack        /tmp/rmconntrack
socket_rmacnt             /tmp/rmacnt
socket_rmpoller           /tmp/rmpoller
pid_dir                   /var/run
cmd_php                   /usr/bin/php
mail_localdomain          localhost.localdomain
php_sendsms               sendsms.php
php_sendmail              sendmail.php
emailwarntraff_tpl        mailwarntraff_tpl.txt
smswarntraff_tpl          smswarntraff_tpl.txt
```

Description of parameters:

- **db_host** – RADIUS database host.
- **db_name** – RADIUS database name.
- **db_user** – RADIUS database user.

- **db_psw** – RADIUS database password.
- **db_host_cts** – CONNTRACK database host.
- **db_name_cts** – CONNTRACK database name.
- **db_user_cts** – Define the CONNTRACK database user.
- **db_psw_cts** – Define the CONNTRACK database password.
- **db_sock** – Define the MySQL socket location.
- **radman_path** – Define the Radius Manager full web path.
- **def_lang** – Default system language (fallback).
- **rootexec_psw** – The password for *rootexec* helper.
- **inactivity** – Timeout in minutes for automatic session cleanup (stale sessions).
- **poller_pause** – Time interval in seconds when *rmpoller* checks the online users and calculates the remaining limits. 60–300 seconds are acceptable. Lower values ensure higher precision in disconnection but generate more system load. Higher values mean less load to system but a slight overconsumption can occur (users can go into negative balance).
- **api_pause** – Mikrotik API cycle pause in seconds
- **cmpoller_pause** – Pause in seconds between two *cmpoller.php* cycles. Enter 60–300 seconds here. Smaller values will ensure more accurate online CM list in ACP.
- **radclient** – Full path of *radclient* binary file.
- **starutil** – Full path of *starutil* binary file.
- **nas_port_mt** – RADIUS POD port for all Mikrotik NAS devices in the system.
- **nas_port_chilli** – RADIUS POD port for all StarOS NAS devices in the system.
- **nas_port_cisco** – RADIUS POD port for all Cisco NAS devices in the system.
- **mt_api_port** – Global API port for Mikrotik.
- **cts_port** – The listener port for syslog messages.
- **cts_blocksize** – CTS data block size.
- **cts_file** – File name of temporary connection storage.
- **cts_threads** – Number of threads for connection data processing.
- **cts_flush** – Flush buffer in every n seconds (default 30 seconds).
- **cts_username_len** – Maximal length of the stored user name in CTS db.
- **cts_allindex** – Create all indexes on CTS tables (use with small tables only).
- **cts_logallip** – Log all IP addresses, not only the authenticated users.
- **socket_rmconntrack** – Rmconntrack server socket.
- **socket_rmacnt** – Rmacnt client socket.
- **socket_rmpoller** – Rmpoller client socket.
- **pid_dir** – Directory of PID files.
- **cmd_php** – Full path of PHP executable.
- **mail_localdomain** – Email local domain.
- **php_sendsms** – SMS sender PHP module.
- **php_sendmail** – Email sender PHP module.
- **emailwarntraff_tpl** – Email template for traffic alert.
- **smswarntraff_tpl** – SMS template for traffic alert.

Radius Manager daemons and utilities

To indetify the issues upon system installation and during the usage it is necessary to understand what Radius Manager components do and how they work? A brief description of Radius Manager executables and utilities is available here.

Binary files

- **rmauth** – Checks the capping, authenticates users, sets bandwidth etc. It is called from *raddb/users.*

- **rmacnt** – Closes the inactive accounting sessions and has other minor functions. Called from *raddb/acct_users*.

- **rmpoller** – This multi function daemon checks the remaining credits (when remote disconnection mode is enabled), disconnects expired users, sends email and SMS alerts, maintains bandwidth on the fly etc. It is a standalone process and should be running all the time.

- **rmconntrack** – Receives Mikrotik syslog messages and stores the CTS data.

- **rootexec** – Used to execute external UNIX programs from PHP. It is essential part of Radius Manager system.

- **rmlic** – License request code generator. Flag **-g** is used to generate the license request code for your system.

PHP files

- **rmscheduler.php** – This module is called daily once by the *cron*. The recommended time for this is some minutes after midnight. It will check the expired RADIUS accounts, unpaid invoices and disables UNIX users. It also does scheduled service changes, disconnects postpaid users on the 1st day of the month to maintain correct postpaid billing period, sends warning emails etc. It is also responsible for auto renewal of accounts.

- **expiryalert.php** – Used for getting data from CTMS and cable modems. It is invoked as a cronjob.

- **newuserscleanup.php** – Used for getting data from CTMS and cable modems. It is invoked as a cronjob.

- **dailyacctcleanup.php** – Prunes the daily accouting table.

- **cmtspoller.php** – Used for getting data from CTMS and cable modems. It is invoked as a cronjob.

- **wlanpoller.php** – Used for getting the wireless client data from APs. It is invoked as a cronjob.

- **phpsesscleanup.php** – Prunes PHP session table cleanup.

These binaries get their configuration from */etc/radiusmanager.cfg* and *config/ system_cfg.php*.

SMS gateway

The SMS gateway is implemented in *smsgateway.php* file. It realizes a simple HTTP / SMS gateway function with **clickatell.com** service. *Smsgateway.php* is a unencoded PHP file. The SMS **gateway credentials** are also defined in this file.

List of functions:

Name:

sendsms

Description:

This function is called when Radius Manager needs to send an SMS message. By default it uses **clickatell.com** gateway. You can also call your own SMS gateway here (a HTTP gateway with CURL or a shell script to use your own mobile phone).

Parameters:

recp – Mobile number.
body – Message body.
errmsg – Pointer to error message returned by the gateway.

Result:

true - API succeeded
false - API error

Remarks:

The function includes a fully implemented **clickatell.com** HTTP / SMS gateway. Any custom SMS gateway can be defined in this function.

Database maintenance

Cumulating the accounting data

With *cumulate.sql* script You can cumulate the accounting data in RADIUS database. The accounting data are stored in the **radacct** table.

Cumulating the accounting data deletes the detailed accounting information from the *radacct* table and creates one accounting record for every user in the selected period. The decreased number of accounting information will speed up the system and reduce the database size.

Complete the following steps to cumulate the accounting information for a certain year:

1. **Enter** the **year** into *cumulate.sql* script.

2. **Execute** *cumulate.sql* script with *mysql* command:

```
[root@localhost]# mysql -u radius -pradius123 radius < cumulate.sql
```

In the example above the MySQL user name is **radius,** the password is **radius123**. Do not insert a space character between the *–p flag* and password.

The script will cumulate the data to December 31. Cumulate the past years only and never the current year.

Pruning the accounting table

You can execute *dbcleanup.sql* script to delete the old accounting data from the RADIUS database.

The steps for deleting the accounting data are:

1. **Enter** the correct **year** in *dbcleanup.sql* script.

2. **Execute** *dbcleanup.sql* with using *mysql* command:

```
[root@localhost]# mysql -u radius -pradius123 radius < dbcleanup.sql
```

In the example above the MySQL user name is **radius,** the password is **radius123**. Do not insert a space character between the *–p flag* and password.

Deleting the accounting data will speed up the system and reduce the database size.

WARNING!

Always **back up the complete RADIUS database** before any database maintenance!

LEGAL NOTE

- **Radius Manager** software and trademark are Copyright © DMA Softlab LLC. All right reserved.
- **ionCube** is Copyright ionCube Ltd.
- **MikroTik** is a registered trademark of MikroTikls corporation.
- **FreeRadius** is Copyright The FreeRADIUS server project. Licensed under GPL.
- **Chillispot** is Copyright Mondru AB. Licensed under GPL.
- **StarOS** is a trademark of Valemount Networks Corporation.
- **MySql** is released under the GNU General Public License.
- **Cisco** is a trademark of Cisco Systems, Inc.