Configuration Guide

FreeRADIUS - PEAP and MSCHAPv2 with LDAP + MySQL + Daloradius web interface for IIT Bombay Wireless Network

INDEX

1. Building Ubuntu 8.04 packages (64 Bit) 2
2. Installing the binary packages 4
3. Configuring the FreeRADIUS with MySQL 5
4. Configuring FreeRADIUS with MySQL and DaloRADIUS 7
5. Setting up LDAP authentication with FreeRADIUS 9
6. Setting up PEAP + MSCHAPv2 authentication with FreeRADIUS 11
7. Configuring Access Point for IITB RADIUS server 12
8. Adding new Access point to RADIUS server 14

1. Building Ubuntu 8.04. packages (64 Bit)

Before building the FreeRadius Ubuntu 8.04 package, we have to apt-get some packages necessary for the build process.

```
sudo su –

apt-get install debhelper libltdl3-dev libpam0g-dev libmysqlclient15-dev build-essential libgdbm-dev libldap2-dev libsasl2-dev libiodbc2-dev libkrb5-dev snmp autotools-dev dpatch libperl-dev libtool dpkg-dev libpq-dev libsnmp-dev libssl-dev
```

Get hold of the FreeRadius sources and start building the package like given below.

```
apt-get source freeradius
```

Note: It is recommended, that you carry this building process on a non-production server & move the final binary packages over to the production server. However, it is not mandatory if you know what you are doing.

```
cd freeradius-1.1.7/
```

Edit rule file `(debian/rules):`

```
vim debian/rules
```

Now search for these lines:

```
--without-rlm_eap_tls \
          --without-rlm_eap_ttls \
          --without-rlm_eap_peap \
```

and change them to look like this.

```
--with-rlm_eap_tls \
          --with-rlm_eap_ttls \
          --with-rlm_eap_peap \
```

Also in the same file replace the text **" --without-openssl "** with **" --with-openssl "**

Now search for these lines and **delete** them.

```
for pkg in $(shell grep ^Package debian/control | awk '{print $$2}') ; d
  o \
    if dh_shlibdeps -p $$pkg -- -O | grep -q libssl; then \
      echo "$$pkg links to openssl" ;\
      exit 1 ;\
    fi ;\
```

```
done
```

Save changes and quit vim.

Now edit the control file `(debian/control):`

Search for the line:

```
Build-Depends: debhelper (>= 5), libltdl3-dev, libpam0g-dev, libmysqlclient15-dev | libmysqlclient-dev, libgdbm-dev, libldap2-dev, libsasl2-dev, libiodbc2-dev, libkrb5-dev, snmp, autotools-dev, dpatch (>= 2), libperl-dev, libtool, dpkg-dev (>= 1.13.19), libpq-dev, libsnmp-dev
```

and append ***libssl-dev*** to the end of this line so that it looks like this.

```
Build-Depends: debhelper (>= 5), libltdl3-dev, libpam0g-dev, libmysqlclient15-dev | libmysqlclient-dev, libgdbm-dev, libldap2-dev, libsasl2-dev, libiodbc2-dev, libkrb5-dev, snmp, autotools-dev, dpatch (>= 2), libperl-dev, libtool, dpkg-dev (>= 1.13.19), libpq-dev, libsnmp-dev, libssl-dev
```

Save the changes and quit vim.

Assuming you are here ~/freeradius-1.1.7. Start building packages:

```
dpkg-buildpackage –r fakeroot
```

Note: You still might require some packages for these. apt-get/aptitiude them & rerun the rebuild process.

After a while (depending on your system) you should have some .deb files in the home directory.

```
freeradius_1.1.7-1build4_i386.deb
freeradius-dbg_1.1.7-1build4_i386.deb
freeradius-dialupadmin_1.1.7-1build4_all.deb
freeradius-iodbc_1.1.7-1build4_i386.deb
freeradius-krb5_1.1.7-1build4_i386.deb
freeradius-ldap_1.1.7-1build4_i386.deb
freeradius-mysql_1.1.7-1build4_i386.deb
freeradius-postgresql_1.1.7-1build4_i386.deb
```

(In this setup **you won't** be needing the postgresql , krb5 , iodbc , dbg, dialupadmin binaries.)

2. Installing the binary packages

Install following packages by typing

```
dpkg –i freeradius_1.1.7-1build4_i386.deb
dpkg -i freeradius-mysql_1.1.7-1build4_i386.deb
dpkg -i freeradius-ldap_1.1.7-1build4_i386.deb
```

After running with the out of the box configuration, validate against a local user.

E.g: run radius in debug mode:

```
freeradius –X
```

From another shell run this while the freeradius -X is running:

```
radtest abc 123 localhost 1812 testing123
```

Make sure the user *abc* with password *123* is set in the `/etc/freeradius/users` file.

3. Configuring the FreeRADIUS with MySQL

First the MySQL bits (creating the db & its admin user). Do the following from your shell.

```
mysqladmin -u root password 123456
mysql -u root -p
```

On the MySQL shell type the following:

```
CREATE DATABASE radius;
GRANT ALL ON radius.* TO radius@localhost IDENTIFIED BY "radpass";
exit;
```

Import the the FreeRadius schema. The sample schema resides at this location:

```
/usr/share/doc/freeradius/examples/mysql.sql.gz.
```

Gunzip it there:

```
gunzip -d /usr/share/doc/freeradius/examples/mysql.sql.gz
```

Do the following:

```
mysql -u root -p radius < /usr/share/doc/freeradius/examples/mysql.sql
```

To have a look at the db schema do the following:

```
mysql -u root -p
use database radius;
show tables;
quit;
```

Now edit your `/etc/freeradius/sql.conf`.

Reset the `user/password/database` parameters to reflect the changes (eg. `radius/radpass/radius`);

To turn the NAS management from MySQL, search for the line

```
readclients = no
```

and change it to

```
readclients = yes
```

Edit the file `/etc/freeradius/radius.conf` and add a line saying `'sql'` to the `authorize{}` section (which is towards the end of the file).

Also add a line saying `'sql'` to the `accounting{}` section to tell FreeRadius to store accounting records in SQL as well.

Optionally add `'sql'` to the `session{}` section if you want to do simultaneous-Use detection. Optionally add `'sql'` to the `post-auth{}` section if you want to log all authentication attempts to SQL.
Here is the authorize section:

```
authorize {
          preprocess
          chap
          mschap
          suffix
          eap
          sql
          pap
}
```

And the accounting section:

```
accounting {
          detail
          sql
 }
```

To insert a test user in the database, go to the MySQL shell and run this:

```
mysql -u root –p
mysql> use database radius;
mysql> INSERT INTO radcheck (UserName, Attribute, Value) VALUES
('sqltest', 'Password', 'testpwd');
mysql> select * from radcheck where UserName='sqltest';
mysql> exit
```

Fire up radius in debug mode:

```
freeradius –X
```

Go to another shell and run the test:

```
radtest sqltest testpwd localhost 1812 testing123
```

At this moment, you should see a message containing something like ... `Accept-Accept` ..., which is an indication that your user is getting authenticated just fine.

Congratulations! **Your FreeRadius + MySQL setup is working.**

4. Setting up web management with Daloradius

The Daloradius latest stable release is version 0.9-7

Get hold of it from http://sourceforge.net/projects/daloradius.

```
tar -zxvf daloradius-0.9-7.tar.gz
cp daloradius-0.9-7/ /var/www –R
```

Download the following prerequisites packages:

```
apt-get install apache2
apt-get install php php-mysql php-pear php-gd php-pear-DB
```

Change permissions and ownership:

```
chown www-data:www-data /var/www/daloradius-0.9-7 -R
chmod 644 /var/www/daloradius-0.9-7/library/daloradius.conf
```

Daloradius needs to add a few more tables to the radius database we already created earlier.

```
mysql -u root -p radius < /var/www/daloradius-0.9-7/contrib/db/mysql-daloradius.sql
```

Now, simply adjust the MySQL database information in the DaloRadius config file.

```
vim /var/www/daloradius-0.9-7/library/daloradius.conf
```

Fill in the database details, a few important parameters are listed below:

```
CONFIG_DB_ENGINE = mysql
CONFIG_DB_HOST = 127.0.0.1
CONFIG_DB_USER = radius
CONFIG_DB_PASS = radpass
CONFIG_DB_NAME = radius
```

Save the file and exit.

Set up the apache server.

Edit the `/etc/apache2/apache2.conf` file and append this to the end of the file (customize to your likings):

```
Alias /myradius "/var/www/daloradius-0.9-7/"
   <Directory /var/www/daloradius-0.9-7/>
      Options None
      order deny,allow
      deny from all
      allow from 127.0.0.1
      allow from <my management system's ip which has a web-browser>
</Directory>
```

Save and exit.
Restart the httpd server:

```
/etc/init.d/apache2 restart
```

Fire up Firefox (or any other borowser) and go to the URL

```
http://<localhost or the managemet system's ip>/myradius
```

Log in with the administrator for management:

```
username: administrator
password: radius
```

Change this information first for the sake of security (info is located in the operator table).

Take Daloradius for a spin. You should have created an sqltest user earlier. You can also try adding new users and testing the connectivity from within the Daloradius frontend.

Congratulations, you are done with **FreeRADIUS + MySQL** setup.

Now we will look at LDAP configuration setting for FreeRADIUS

5. Setting up LDAP authentication with FreeRADIUS

Open `/etc/freeradius/radius.conf` and search for

```
#ldap {
      #server=
```

Modify it with IIT Bombay specific LDAP configuration

```
ldap {
            server = "ldap.iitb.ac.in"
            identity = "cn=USERNAME,ou=people,dc=iitb,dc=ac,dc=in"
            password = PASSWORD
            basedn = "dc=iitb,dc=ac,dc=in"

            #filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
            # base_filter = "(objectclass=radiusprofile)"

            filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
            base_filter   = "(objectclass=posixAccount)"

            # set this to 'yes' to use TLS encrypted connections
            # to the LDAP database by using the StartTLS extended
            # operation.
                              :
                              :
                              :
             set_auth_type = yes
      }
```

Where USERNAME = user having read access to LDAP database
Password = password of that user (without " ")

Now, search for

```
# Uncomment it if you want to use ldap for authentication
      #
      # Note that this means "check plain-text password against
      # the ldap database", which means that EAP won't work,
      # as it does not supply a plain-text password.
      #Auth-Type LDAP {
      #     ldap
      #}
```

and uncomment it

```
      Auth-Type LDAP {
            ldap
      }
```

Search for

```
#
#  The ldap module will set Auth-Type to LDAP if it has not
#  already been set
#ldap
```

and uncomment it

```
#
#  The ldap module will set Auth-Type to LDAP if it has not
#  already been set
ldap
```

Now, open `/etc/freeradius/ldap.attrmap` and uncomment all lines (remove "#" from all lines)

```
checkItem  $GENERIC$             radiusCheckItem
replyItem  $GENERIC$             radiusReplyItem
checkItem  Auth-Type             radiusAuthType
checkItem  Simultaneous-Use      radiusSimultaneousUse
checkItem  Called-Station-Id     radiusCalledStationId
checkItem  Calling-Station-Id    radiusCallingStationId
checkItem  LM-Password           lmPassword
checkItem  NT-Password           ntPassword
checkItem  SMB-Account-CTRL-TEXT acctFlags
                      :
                      :
replyItem  Reply-Message         radiusReplyMessage
```

6. Setting up PEAP + MSCHAPv2 authentication with FreeRADIUS

Open `/etc/freeradius/eap.conf` and search for `eap{`

Make sure `default_type = peap` as shows below

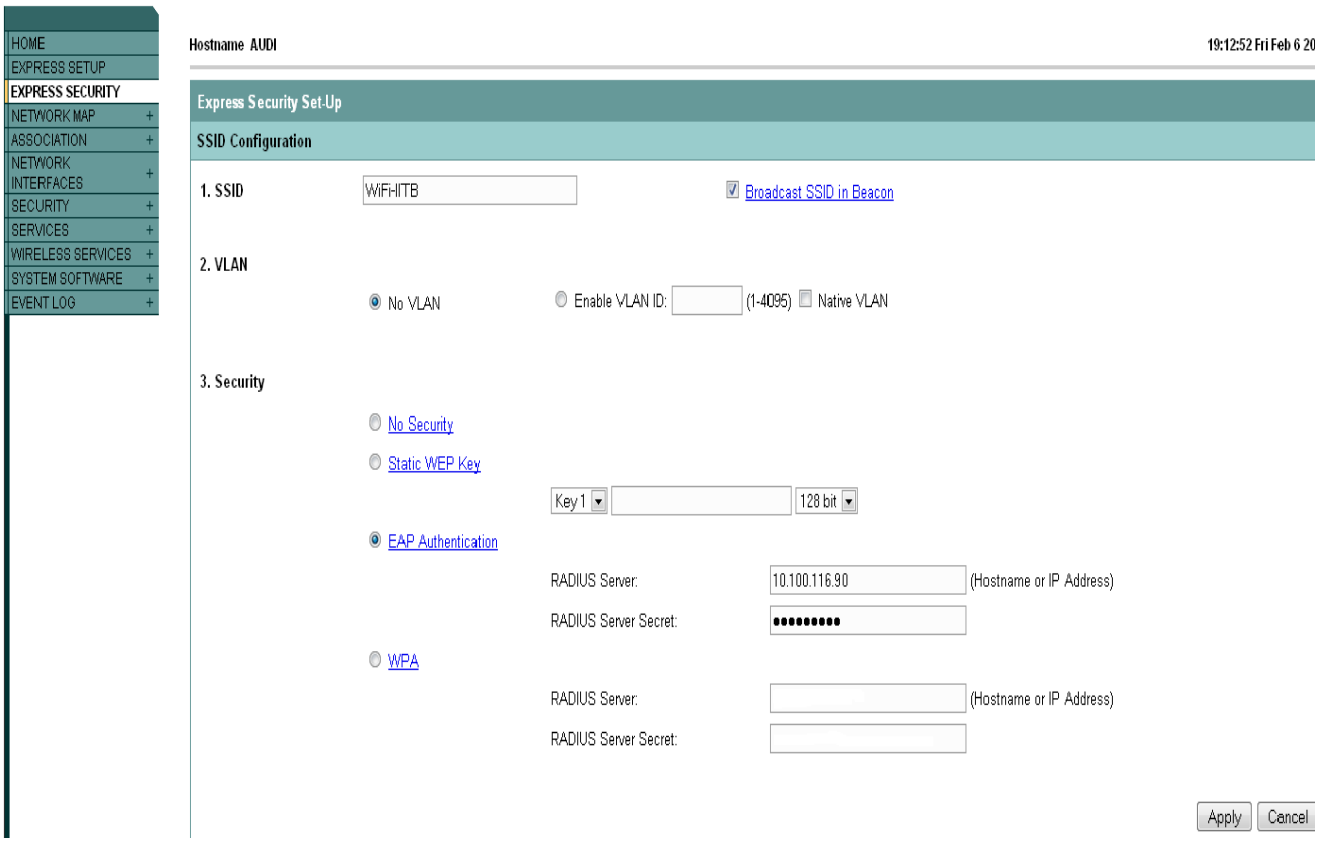
```
eap {
            #  Invoke the default supported EAP type when
            #  EAP-Identity response is received.
            #
            #  The incoming EAP messages DO NOT specify which EAP
            #  type they will be using, so it MUST be set here.
            #
            #  For now, only one default EAP type may be used at a
time.
            #
            #  If the EAP-Type attribute is set by another module,
            #  then that EAP type takes precedence over the
            #  default type configured here.
            #
            default_eap_type = peap
                                    :
                                    :
```

That's it! Now USER should able to login using LDAP login once access point is configured to authenticate using RADIUS server.

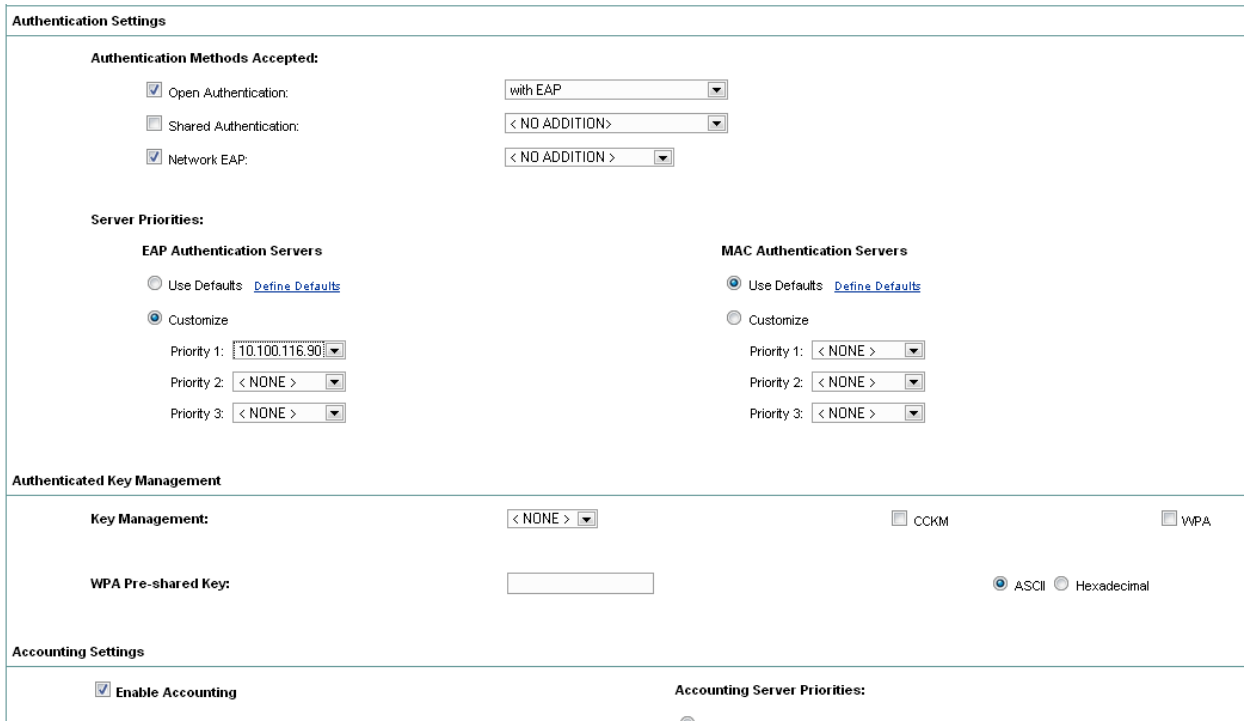
7. Configuring Access Point for IITB RADIUS server

We are assuming RADIUS server IP as **10.100.116.90**

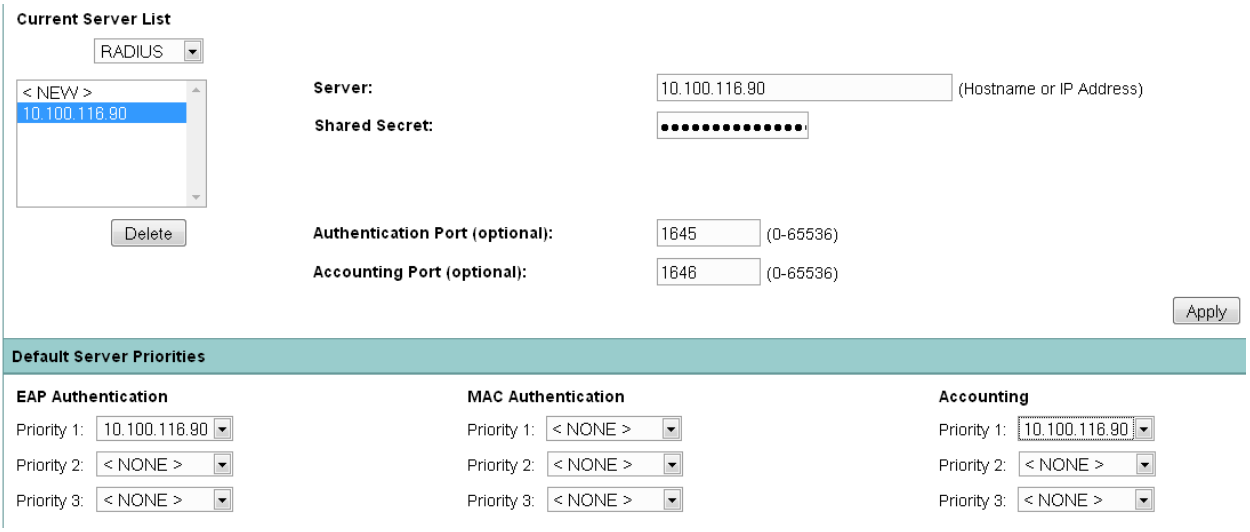
A. Click on **Express Security** on left panel and do following setting and click on **apply**



B. Click on **SECURITY → SSID Manager** and do following setting and click on **apply**



C. Click on **SECURITY → SERVER Manager** and do following setting and click on **apply**



8. Adding new access point to RADIUS

Once access point is configured to use specific RADIUS server, we need to add its IP address to RADIUS server database.

To add new Access point, open `/etc/freeradius/clients.conf` and add following entry at the end of file (next new entry will be appended to current one)

```
client 10.99.32.226 {
      secret     = SET_IN_ACCESS_POINT
      shortname  = cisco
}
```

Where **SET_IN_ACCESS_POINT** is the secret that you had entered while configuring access point and **10.99.32.226** is an IP address of access point.

Note: You have to make entry for each access point.

You are now ready to deploy RADIUS server with integrated LDAP and MySQL authentication. (For adding user accounting using SQL counter refer to http://wiki.freeradius.org/SQL_HOWTO)

Good Luck!

References

[1] http://freeradius.org/

[2] http://wiki.freeradius.org/SQL_HOWTO

[3] http://sourceforge.net/projects/daloradius

[4] http://www.howtoforge.com/wifi-authentication-accounting-with-freeradius-on-centos5

[5] http://www.linuxinsight.com/building-debian-freeradius-package-with-eap-tls-ttls-peap-support.html

[6] http://www.howtoforge.com/setting-up-a-freeradius-based-aaa-server-with-mysql-and-management-with-daloradius