Instructions for using FreeRADIUS for Wi-Fi authentication - Part 1

We can use the FreeRADIUS server as a user authentication server, which is an open source project, developed in the GNU General Public License Version 2 (GPLv2). It is indeed a RADIUS server that has been used all over the world.

Network administration - **Wireless networks for businesses, large or small, must always be protected by the enterprise mode of Wi-Fi Protected Access (WPA or WPA2). This is a mode that provides stronger encryption to protect your network against Wi-Fi attacks.** It can also hide encryption keys for your users, so users will be more difficult to compromise, unlike mode (PSK) (personal or pre-shared key) of WPA or WPA2. , the distribution of encryption keys is vulnerable to attack, because users can disclose their keys to malicious code programs, although not intentionally.

However, the enterprise version requires you to use a RADIUS server. This server will help users to authenticate themselves to be able to access the network. Instead of entering the encryption key, the user logs in to the network with the username and password. Real keys will be exchanged without the user knowing and the user's key is different as well as being updated regularly.

We can use the FreeRADIUS server as a user authentication server, which is an open source project, developed in the GNU General Public License Version 2 (GPLv2). It is indeed a RADIUS server that has been used all over the world. In addition to performing 802.1X / PEAP authentication, the authentication we will set up, this server also supports many other types of authentication for a variety of network types. It also has the ability to automate failover, load balancing and support multiple backend databases.

First you need to install a Linux distribution. This tutorial is based on CentOS operating system, a free open source operating system. Besides Mac OS X and Windows operating systems are also supported, as well as Linux distributions.

Tip *: If you do not perform a new default installation of CentOS 5.3, you need to make sure you have installed the OpenSSL package before proceeding.*

You can install FreeRADIUS on any old PC. Just make sure that the FreeRADIUS computer has a wired connection to the network. In addition, make sure that it has a static IP address rather than a dynamic address. Assign this address to the network adapter in CentOS or reserve another address through the router's DHCP settings.

Note *: Installation instructions are based on version 5.3 of CentOS operating system and FreeRADIUS version 2.1.6. Now the current FreeRADIUS packages are not provided through the usual CentOS hosting address, but only the older version 1.xx packages. So we will use a third-party location. However, you can still install the current version (2.xx) of FreeRADIUS by using software packages via Package Manager or in some way.*

1. Using CentOS, download the freeradius2.repo file and save it to the desktop.
2. Open a **Terminal** and type " **su** ", enter your root password. Then type " **cp / home /** ***yourusername*** **/Desktop/freeradius2.repo /etc/yum.repos.d** ".
3. Continue typing " **yum install freeradius2** ", and when prompted, enter " **y** " to begin the installation process.
4. If dependencies are needed, choose to install them.

Tip *: If you get a "Package is not signed" error, type "gedit" and use the text editor to change "gpgcheck = 1" to "gpgcheck = 0" in the / etc / yum file .conf, then save and close the text editor. After installation is complete, reverse these settings. Enter the installation line again in an existing terminal window.*

You can install additional FreeRADIUS packages, such as a backend database support package. To see the list of packages, type " **yum info freeradius2 *** ". In this tutorial, we only use MySQL, so we installed it with the command " **yum install freeradius2-mysql** ". Next, choose to install dependencies by typing ' **y** '.

Start the configuration file

If you have not worked with Unix / Linux servers or command line applications, FreeRADIUS can be quite confusing for you now. Although there are several GUI utilities available, it is usually configured through configuration text files.

Installing FreeRADIUS is very simple. The default configuration files are pre-configured to run most authentication protocols without much or any changes.

Do not change or delete settings without understanding what it is and what it does. A simple error can damage the configuration and may take a long time to fix the problem. If you have any changes outside of this tutorial, take steps one by one. Change a setting or part containing settings, then test what happened to your changes.

Create self-signed certificates for PEAP

Although SSL certificates are required for PEAP, TLS is now created automatically by FreeRADIUS, so you still have to customize the identity and password attributes. Do this before running the server for the first time and this is how to make the changes.

1. Open a **terminal** , type " **su** " for root mode, and run " **gedit** " to open the text editor. Then open **ca, client** and **server files cnf** from **/ etc / raddb / certs** . In each configuration file, edit the following:
    1. Change " *default_days* " in the CA Default section to a number greater than one year, so you don't have to create and upgrade the certificate right away.
    2. Change " *input_password* " and " *output_password* " in the **Req** section so that the certificates are protected instead of the default password.
    3. Change 6 values ??for identity fields in the **Certificate Authority** , **Client,** and **Server** sections.
2. Save files but don't close the editor at this time.
3. You need to upgrade the password in the **etc / raddb / eap.conf file** by changing the " *private_key_password* " value in the TLS section.
4. Save the file and close the editor.

In the existing root terminal, type **"/ usr / sbin / radiusd -X** ". This command will create your self-signed certificates and start the server in debug mode so you can see what is happening. If everything goes as planned, you will see the words " **Ready to process requests** " last.

Although the server is now installed and can be run, the following sections will show you how to configure some other settings before you are ready to authenticate your Wi-Fi users.

Install EAP settings

There are many types of EAP, so you must specify which type you want to use. In this tutorial we will discuss the use of PEAP, an EAP type that does not require you to create security certificates for each user. They connect and network using their username and password.

When you are ready, make a small change to the EAP configuration file:

1. Open **Terminal** , type " **su** " for root mode, and run " **gedit** " to open the Text Editor. Then open ***etc / raddb / eap.conf*** .
2. In the first part of the **EAP** section, change the " *default_eap_type* " from " *md5* " to " *peap* ".
3. Save and close the file, but still open the Text Editor.

Create user account

Next you need to create usernames and passwords that users will enter when connecting to Wi-Fi networks. First we will create a user account in the configuration file to test the server. We will then use the MySQL database to store user information, which is a great solution if you have multiple users or need to change their password information on a regular basis.

In the existing root text editor, open *etc / raddb / users* . Then type username, press Tab and type Cleartext-Password: = " *thepassword* ".

Here is an example:

egeier Cleartext-Password: = "pass123"

Save and close the file, but still open the editor.

Enter the details of the AP (client)

Now you have to enter the shared IP address and secret (password) of at least one wireless AP, which is called the client for FreeRADIUS. Next you can store client details in a database, such as MySQL. However, if you are working on a small network, you can easily use the text file method.

On an existing root text editor, open *etc / raddb / clients.conf* and enter details somewhere for each AP according to the following example:

```
client 192.168.0.1 {
secret = testing123
shortname = private-network-1
}
```

Change the IP address, enter a unique secret for each AP and can enter a descriptive name. After that, don't forget to save the file when you're done making the changes.