This document describes the steps to install freeRadius under Fedora and prepare configuration to be used to authenticate PacketShaper Login Access.

I do have Fedora with kernel 2.6.9-1.667 running and downloaded a copy of freeRadius 2.1.7 from their website http://freeradius.org

Make sure you have Development Tools installed on your Fedora Workstation

A comprehensive Source of Installation & Configuration is found on http://wiki.freeradius.org/Main_Page

Unpack the download gz file

```
[root@fedora ~]# gunzip freeradius-server-2.1.7.tar.gz
```

Extract tar file

```
[root@fedora ~]# tar -xvf freeradius-server-2.1.7.tar
```

Simple Installation of freeRadius

You will need to prepare the package of freeRadius to get it installed on your own dedicated linux workstation. To to that, just run following three commands. Don't get surprised, you will see a lot of messages running on the screen when running those commands. Make install will add all required files to your system to be ready to run freeRadius the first time.

```
[root@fedora freeradius-server-2.1.7]# ./configure
[root@fedora freeradius-server-2.1.7]# make
[root@fedora freeradius-server-2.1.7]# make install
```

The first time, you should start the freeRadius Server under root. The 'X' will run the server in debugging mode. The will also generate a Certificate.

```
[root@fedora freeradius-server-2.1.7]# radiusd -X
```

If you see following messages at the end your newly installed freeRadius is listening to the requests.

Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on command file /usr/local/var/run/radius/radius.sock
Listening on proxy address * port 1814

However, this is only half of the story. Next you will need to configure the Radius Server to respond with the vendor specific access level attribute.

Configure Radius Server

1. Create a file named dictionary.packeteer

Create a file named **dictionary.packeteer** (typically in the /usr/local/share/freeradius) with these lines, if not present already. In this installation with freeRadius Version 2.7.1, dictionary.packeteer was already included.

```
VENDOR          Packeteer                       2334

#
#       Standard attribute
#
BEGIN-VENDOR    Packeteer

ATTRIBUTE       Packeteer-AVPair                        1       string

END-VENDOR      Packeteer
```

2. Add the following line to the file named dictionary

Even the PacketGuide (User Manual of a PacketShaper) stated, you have to add following line ***"$INCLUDE dictionary.packeteer"*** to the dictionary file, there is no need to do so, because the existing include statement points to a directory with all vendor specific dictionary files.

```
$INCLUDE      /usr/local/share/freeradius/dictionary
```

3. Enter each user's name, password, and local access level into the users file

Copy the original ***users*** (/usr/local/etc(raddb) file to and create an empty file. Personally, I prefer slim files with only some lines. I also did use user names from the local fedora workstation, to control passwords from a system point of view. However, keep in mind, only PAP will work. I did use PacketWise 8.3.3. It may change in the future, but that question needs to get to BlueCoat, who owns PacketShaper after their acquisition of Packeteer.

```
rbemsel Auth-Type := System
        Packeteer-AVPair = "access=look"
root Auth-Type := System
        Packeteer-AVPair = "access=touch"
```

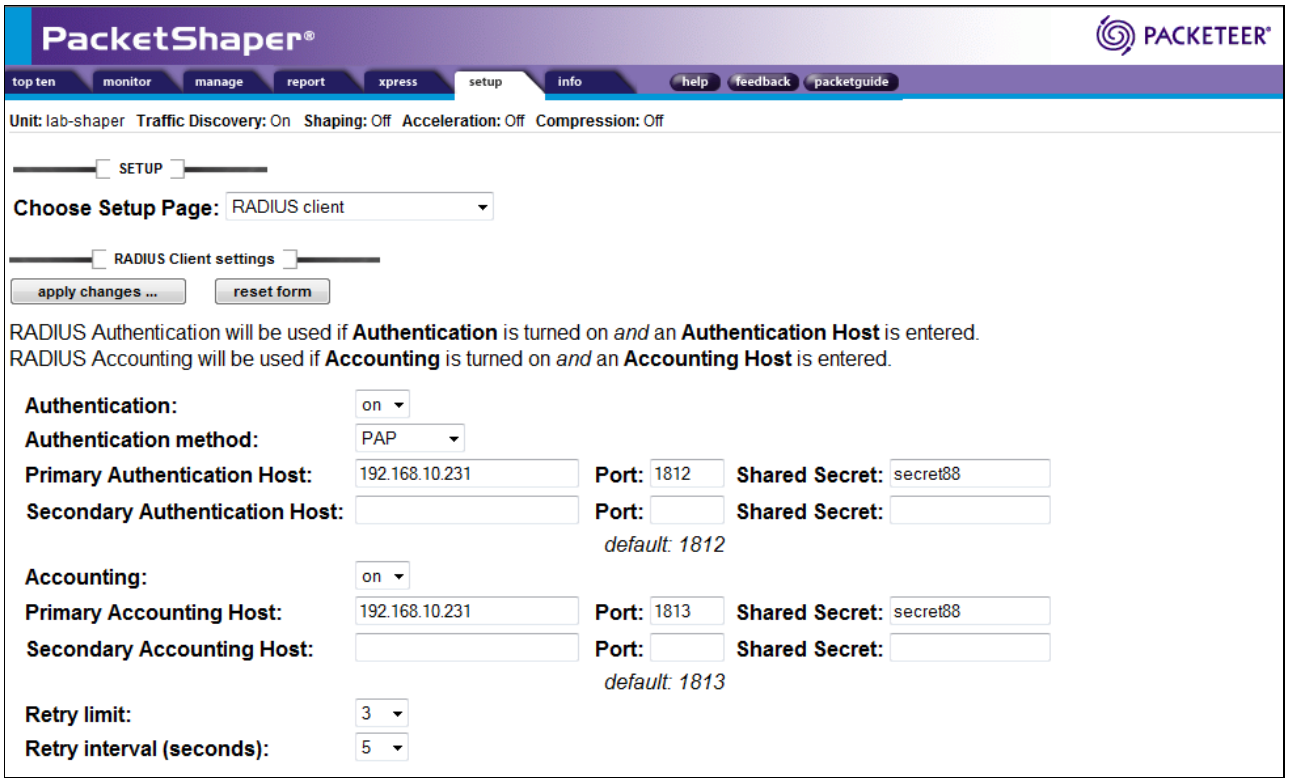
4. Add Client IP and dedicated Shared Secret

Edit the client.conf (/usr/local/etc(raddb) and add following lines with the PacketShaper IP Address and correlated Shared Secret. My PacketShaper is using 192.168.10.83

```
#}

client 192.168.10.83 {
        secret          = secret88
        shortname       = lab-shaper
}
```

Configure Radius Authentication Service on PacketShaper

Log into a PacketShaper



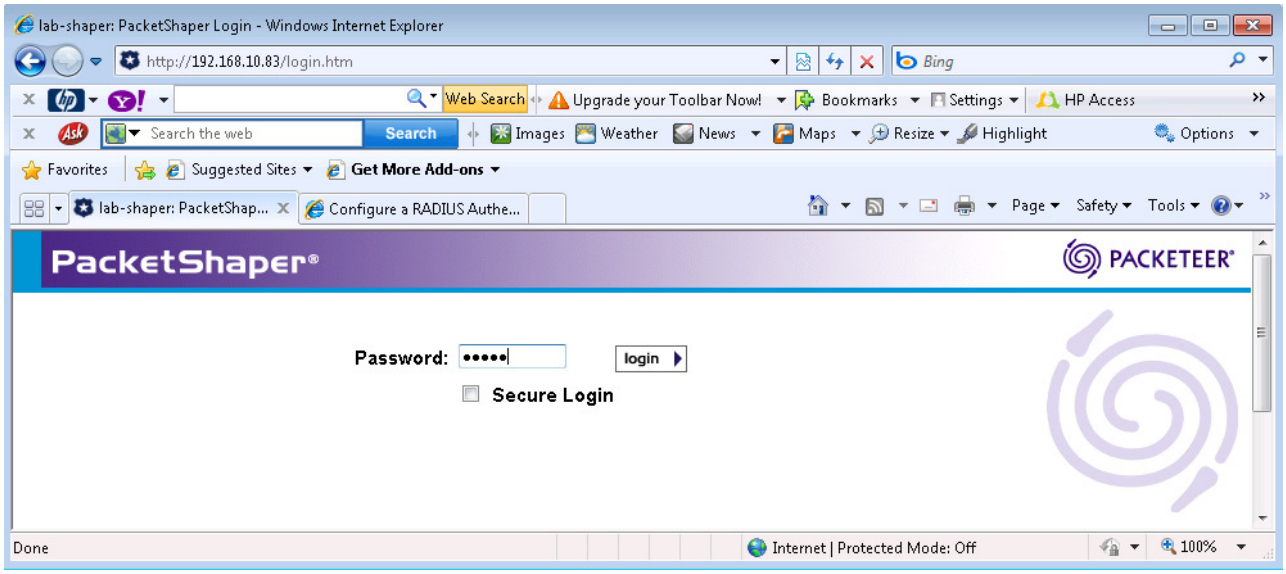
You can get a more detailed response, when loggin into the console or telnet/ssh to the CLI

```
192.168.10.83# radius session

ID       Status         Age        Idle      Limit    Type Access User Name
-------------------------------------------------------------------------------
4b1c20af logged in    125 secs     0 secs    60 mins CLI  touch  root
4b1c2021 logged in    260 secs   202 secs    60 mins WUI  touch  root

192.168.10.83#
```

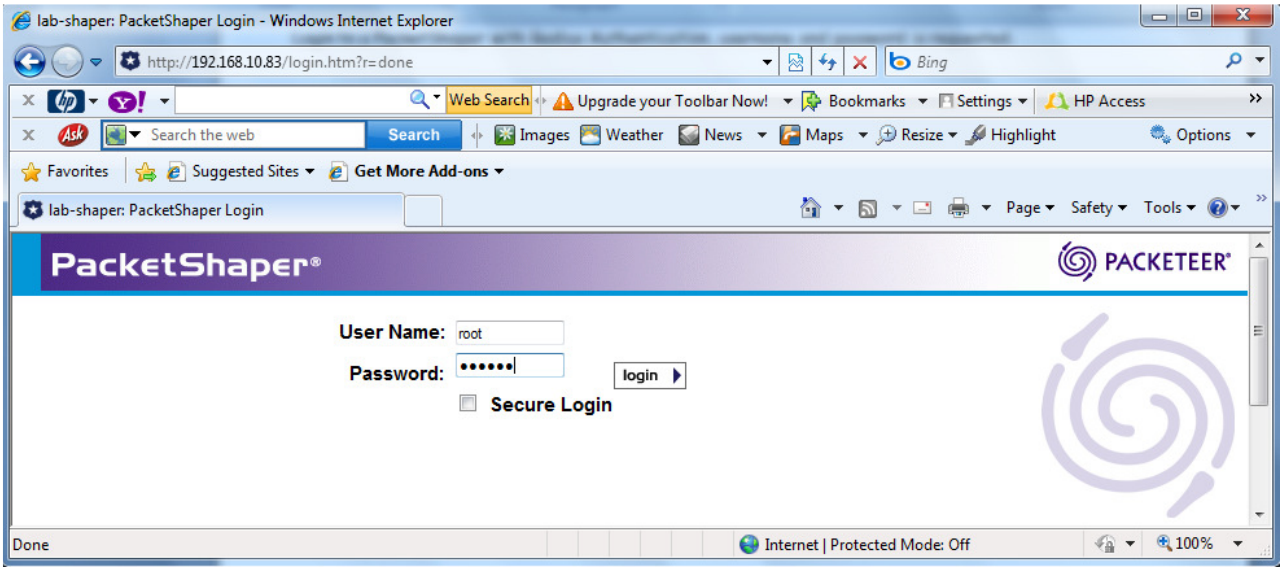
I like following login test, as I can see immediate response.

```
192.168.10.83# radius login root my_secret_password
"root" RADIUS Authentication OK
Vendor-Specific: access=touch

192.168.10.83#
```

Login to a PacketShaper with Standard Authentication, only password is required.

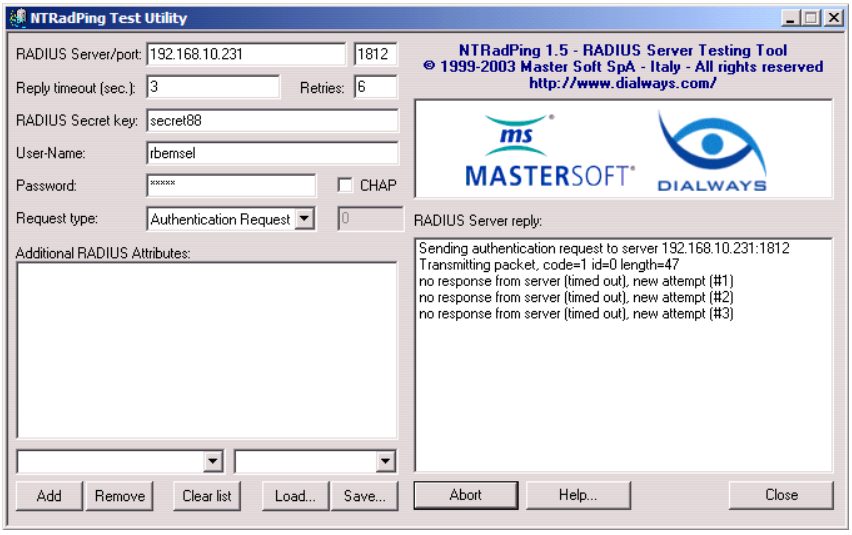


Login to a PacketShaper with Radius Authentication, username and password is required.

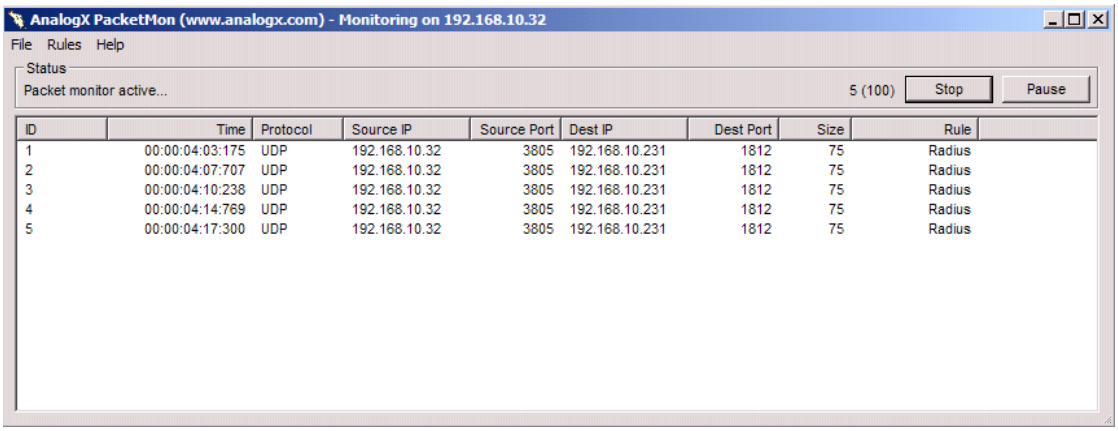


Troubleshooting Tips

This is a free Test Utility, which can be downloaded at www.dialways.com . Using this tool does not require any other radius clients to connect to the server.



Additionally, I use to Packet Monitoring Tool, which can be downloaded at www.analogx.com. This is also freeware and helps to determine if packets are leaving and receiving correct. Easy to define filters. It does not capture data, but show incoming and outgoing connections. Very useful



ID	Time	Protocol	Source IP	Source Port	Dest IP	Dest Port	Size	Rule
1	00:00:04:03:175	UDP	192.168.10.32	3805	192.168.10.231	1812	75	Radius
2	00:00:04:07:707	UDP	192.168.10.32	3805	192.168.10.231	1812	75	Radius
3	00:00:04:10:238	UDP	192.168.10.32	3805	192.168.10.231	1812	75	Radius
4	00:00:04:14:769	UDP	192.168.10.32	3805	192.168.10.231	1812	75	Radius
5	00:00:04:17:300	UDP	192.168.10.32	3805	192.168.10.231	1812	75	Radius

Finally, if you have started freeRadius in debugging mode, there is a lot of information, how authentication using the radius protocol works

Don't forget to use tcpdump on Linux to see if Radius Packets are received

22:32:11.379864 IP 192.168.10.83.1088 > 192.168.10.231.radius: RADIUS, Access Request (1), id: 0x91 length: 59
22:32:11.394389 IP 192.168.10.231.radius > 192.168.10.83.1088: RADIUS, Access Accept (2), id: 0x91 length: 40