Master Thesis

Installing and Configuring Security Mechanisms
Freeradius-MySQL
Freeradius-LDAP
PAM/USB Modules
LinOTP

Niskopoulos Nikolaos
Student No: MTE 1058
Supervising Professor: Xenakis Christos

Piraeus
March 2012
This thesis is dedicated to my parents who supported all my choices and efforts
Acknowledgements

My sincere thanks to Assistant Professor Christos Xenakis and the Postdoctoral researcher Christoforos Ntantogian, my project supervisors, for their valuable input, guidance and support.

Thanks to all the lecturers and students at the Postgraduate Program for sharing their knowledge.

Special thanks to my family and friends for their patience, ongoing support and encouragement throughout the years.
Abstract

This master thesis is a compilation of instructions-“how to” guides in order to install and configure security mechanisms that are of crucial importance considering the numerous threats any system administrator has to confront during his daily obligations.

The first security mechanism that has been installed and configured was Freeradius combined with MySQL database in Ubuntu 11.10 operating system. The freeradius was installed to a local network and wan configured to authenticate users stored to the SQL database via PAP (Password Authentication Protocol) and EAP-TLS (Extended Authentication Protocol-Transport Layer Security) by using certificates created using the free OpenSSL tool.

The second security mechanism was again Freeradius only this time was supported by an LDAP database in Ubuntu 11.10 operating system. The Lightweight Directory Access Protocol (LDAP) is an open standard for accessing directory services, X.500. The protocol runs over transport layer (OSI) where in the internet case is TCP/IP. The directory service is a database that organizes records and improves the procedures of accessing and searching. In this thesis we configure the LDAP database via the web user interface phpldapandmin and JXplorer tool in order to create groups and users who authenticate to our network through freeradius.

Furthermore a PAM/USB module was installed in order to authenticate to the Operating System (which was once again Ubuntu 11.10) via USB flash drive without the need of a password. Two factor authentication was also implemented to the OS by using something we possess (USB drive) and something we know (Password).

Finally the LinOTP2 tool was installed and configured in order to authenticate to our operating system via One Time Password. The LinOTP server and an application to our mobile which generates the one time passwords are used to achieve the OTP authentication. Additionally we implement two-factor authentication by using a combined password of something that we know and something that we generate real time. The last configuration is a triple factor authentication using the previous two factors (password and phone OTP) in addition to the PAM/USB module that was installed in the third part of this Thesis.
## Contents

Acknowledgements ................................................................................................................... 3  
Abstract ..................................................................................................................................... 4  
Contents .................................................................................................................................... 5  
Picture Table .............................................................................................................................. 7  
Freeradius Using MySQL on Linux Ubuntu 11.10 .................................................................... 10  
  Introduction .......................................................................................................................... 10  
  Installing FreeRadius Server .............................................................................................. 10  
  Configuring the Freeradius Server ....................................................................................... 21  
  Testing our configuration .................................................................................................... 25  
  802.1X with EAP-TLS Authentication ................................................................................... 28  
  How to create the Certificates .............................................................................................. 28  
  Freeradius configuration for EAP-TLS ............................................................................... 33  
  Radius Server Response ........................................................................................................ 42  
Freeradius with LDAP on Linux Ubuntu 11.10 ......................................................................... 46  
Freeradius with LDAP .............................................................................................................. 46  
  Installing Freeradius with LDAP .......................................................................................... 47  
  Now we start the freeradius configuration ......................................................................... 48  
  Installing phpladapadmin ..................................................................................................... 50  
  NTRadping Test Utility ........................................................................................................ 51  
  Configuring the LDAP Directory ......................................................................................... 52  
  Using NTRadping to test if all the configuration is correct ............................................... 56  
Creating Users by using JXplorer tool ..................................................................................... 60  
PAM/USB Module ....................................................................................................................... 69  
  Introduction .......................................................................................................................... 69  
  Installing PAM/USB Module ................................................................................................. 69  
  Two-Factor Authentication ................................................................................................. 74  
LinOTP tool ............................................................................................................................... 76  
  Introduction .......................................................................................................................... 76  
  Installing the linOTP2 tool ................................................................................................. 76  
  Configuration ......................................................................................................................... 81  
  Enroll OTP Token ................................................................................................................ 83
TWO factor Authentication ............................................................... 87
Authenticate to our Operating System Desktop ........................................ 88
Triple factor authentication ................................................................. 90
References ....................................................................................... 93
Freeradius MySQL ............................................................................. 93
Freeradius LDAP ................................................................................ 93
PAM/USB Module ............................................................................. 94
LinOTP ......................................................................................... 94
**Picture Table**

<table>
<thead>
<tr>
<th>Picture</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Radacct table</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://localhost/phpmyadmin">http://localhost/phpmyadmin</a></td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Creating the tables in SQL tab of phpMyAdmin</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>NAS table</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>The Radius database on phpMyAdmin</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Default configuration file</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Default configuration file</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>Default configuration file</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Default configuration file</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>radius.conf configuration file</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>radius.conf configuration file</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>radius.conf configuration file</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>radius.conf configuration file</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>radius.conf configuration file</td>
<td>18</td>
</tr>
<tr>
<td>15</td>
<td>sql.conf configuration file</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>sql.conf configuration file</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>inner-tunnel configuration file</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>The radius server is up and running waiting for requests</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>Creating the radius clienets on NAS</td>
<td>21</td>
</tr>
<tr>
<td>20</td>
<td>Creating the users of our radius database</td>
<td>22</td>
</tr>
<tr>
<td>21</td>
<td>The users in our freeradius-mysql server</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>configuring the groups of our freeradius-mysql server</td>
<td>23</td>
</tr>
<tr>
<td>23</td>
<td>the existing groups of the freeradius server</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>configuring the user attributes</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>The configuration for user Nisk Niskopoulos</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>configuring the authentication method</td>
<td>25</td>
</tr>
<tr>
<td>27</td>
<td>Configuring the Framed-Protocol the group users</td>
<td>25</td>
</tr>
<tr>
<td>28</td>
<td>Creating Client NTRadping Test</td>
<td>26</td>
</tr>
<tr>
<td>29</td>
<td>Creating user ping</td>
<td>26</td>
</tr>
<tr>
<td>30</td>
<td>Creating group radping</td>
<td>26</td>
</tr>
<tr>
<td>31</td>
<td>Setting the authentication type for group radping</td>
<td>27</td>
</tr>
<tr>
<td>32</td>
<td>Configuring the radping group</td>
<td>27</td>
</tr>
<tr>
<td>33</td>
<td>Configuring user ping</td>
<td>27</td>
</tr>
<tr>
<td>34</td>
<td>The NTRadtest Utility</td>
<td>28</td>
</tr>
<tr>
<td>35</td>
<td>The Pirelli router setup web page</td>
<td>34</td>
</tr>
<tr>
<td>36</td>
<td>How to install certification authority to windows client</td>
<td>37</td>
</tr>
<tr>
<td>37</td>
<td>How to install user certificate to windows 7</td>
<td>38</td>
</tr>
<tr>
<td>38</td>
<td>How to install user certificate to windows 7</td>
<td>39</td>
</tr>
<tr>
<td>39</td>
<td>Configuring windows 7 wireless network center</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>Configuring wireless 802.1x settings</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>Windows network configuration screenshot</td>
<td>40</td>
</tr>
</tbody>
</table>
Creating user1

Create new user account

Creating new child entry

Posix Group adminuser

Generic Posix Group

Creating an organizational unit

Creating LDAP entry

Creating an organizational unit

Login to LDAP Directory

Testing the curent configuration with NTRadPing Test utility

Freeradius server screenshot

Connection to wireless client ONTelecoms

Selecting the preferable certificate

Insert admin credentials

Confirm Security Exception

Confirm user1

Password Request

Switch User Account

Common-auth Configuration File

Adding a user to device PAMUSB-1

The JXplorer Tool

Access Accept with NTRadtest Utility

Successful authentication via usb TRAVELDRIVE

Authentication Failure without usb flash drive

Renter Password while usb is plugged in

Installing Linotp2

Package Installer

Add Exception

Confirm Security Exception

Insert admin credentials

LinOTP2 WEB User Interface

Terminal Screenshot

Users List

Package Installer

Add Exception

Creating user1

Create new user account

Creating new child entry

Posix Group adminuser

Generic Posix Group

Creating an organizational unit

Creating LDAP entry

Creating an organizational unit

Login to LDAP Directory

Testing the curent configuration with NTRadPing Test utility

Freeradius server screenshot

Connection to wireless client ONTelecoms

Selecting the preferable certificate

Insert admin credentials

Confirm Security Exception

Confirm user1

Password Request

Switch User Account

Common-auth Configuration File

Adding a user to device PAMUSB-1

The JXplorer Tool

Access Accept with NTRadtest Utility

Successful authentication via usb TRAVELDRIVE

Authentication Failure without usb flash drive

Renter Password while usb is plugged in

Installing Linotp2

Package Installer

Add Exception

Confirm Security Exception

Insert admin credentials

LinOTP2 WEB User Interface

Terminal Screenshot

Users List
Picture 88: Main Page of android application MOTP .............................................................. 83
Picture 89: Insert Pin and Secret Value ................................................................................... 84
Picture 90: Generate a OTP ..................................................................................................... 84
Picture 91: The OTP is generated ............................................................................................. 84
Picture 92: MOTP passcode generator .................................................................................... 85
Picture 93: Create a token to our Server .................................................................................. 85
Picture 94: mOTP Token ........................................................................................................... 86
Picture 95: New server Token .................................................................................................. 86
Picture 96: LinOTP test page ................................................................................................... 87
Picture 97: Successful authentication ...................................................................................... 87
Picture 98: OTP request ........................................................................................................... 89
Picture 99: OS Desktop ............................................................................................................ 89
Picture 100: Common-linotp configuration file ....................................................................... 91
Picture 101: Permission Denied .............................................................................................. 91
Picture 102: Login Successfully to OS Desktop ......................................................................... 92
Freeradius Using MySQL on Linux Ubuntu 11.04

Introduction

FreeRADIUS is a modular, high performance free RADIUS suite developed and distributed under the GNU General Public License, and is free for download and use. The FreeRADIUS Suite includes a RADIUS server, a BSD-licensed RADIUS client library, a PAM library, an Apache module, and numerous additional RADIUS related utilities and development libraries. In most cases, the word "FreeRADIUS" refers to the free open source RADIUS server from this suite.

FreeRADIUS is the most popular open source RADIUS server and the most widely deployed RADIUS server in the world. It supports all common authentication protocols, and the server comes with a PHP-based web user administration tool.

Installing FreeRadius Server

We start by installing Freeradius on Ubuntu and do some basic testings:

1) Update our apt-get tool

```
sudo apt-get update
```

2) Install the tasksel tool in order to install lamp-server later on

```
apt-get install tasksel
```

3) Install lamp-server

Installing MySQL and PHP are all we need but to make sure that there are no dependencies errors later, we will install the whole package (lamp-server stands for Linux-Apache-MySQL-PHP server).

```
sudo tasksel install lamp-server
```

(you will need to enter root password, which is “setupRADIUS” by default, to continue the installation)
4) **Install phpmyadmin**

We run the following command:

```bash
sudo apt-get install phpmyadmin
```

Once phpMyAdmin is installed point your browser to [http://localhost/phpmyadmin](http://localhost/phpmyadmin) to start using it. You should be able to login using any users you've setup in MySQL during install. If no users have been setup, use `admin` with no password to login.

Should you get a 404 "Not Found" error when you point your browser to the location of phpMyAdmin (such as: [http://localhost/phpmyadmin](http://localhost/phpmyadmin)) this is likely caused by not checking the 'Apache 2' selection during installation.

To redo the installation run the following:

```bash
sudo dpkg-reconfigure -plow phpmyadmin
```

Then select Apache 2 for the webserver you wish to configure. *(we select the apache 2 using space bar not enter! a red dot will appear when apache is selected)*

If this does not work, then you can do the following to include the phpMyadmin-shipped Apache configuration into Apache:

```bash
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
sudo /etc/init.d/apache2 reload
```

5) **Install freeradius package**

```bash
sudo apt-get install freeradius
```

*(Enter password when asked. When you are asked to continue, type ‘y’, then press Enter)*

6) **Install freeradius with ldap authentication**

```bash
sudo apt-get install freeradius-ldap
```

7) **Install freeradius to run with mysql**

```bash
sudo apt-get install freeradius-mysql
```

8) **After finishing the above installations, restart the FreeRADIUS service**

```bash
sudo /etc/init.d/freeradius restart
```
9) Now, you can test the Radius Server using radtest, the command will be as below:

```
nike@ubuntu:~$ radtest radius setupRADIUS localhost 1812 testing123
```

At this stage, you might get an error as follow:

```
radclient: socket: cannot initialize udpfromto: Function not implemented
```

If that’s the case, in order to fix the error, you open /etc/hosts as root, then comment out the line below:

```
::1 localhost ip6-localhost ip6-loopback
```

The correct result should be as follow:

```
Sending Access-Request of id 177 to 127.0.0.1 port 1812
User-Name = "radius"
User-Password = "setupRADIUS"
NAS-IP-Address = 127.0.0.1
NAS-Port = 1812
rad recv: Access-Accept packet from host 127.0.0.1 port 1812, id=177, length=28
```

10) Now, we create the database for FreeRADIUS and a user account which will be used by FreeRADIUS to access into database. We run the following command:

- Login MySQL Database

```
nike@ubuntu:~$ mysql -u root -p
Enter password: *
```

(Enter the password for root user)

- Then you can create user, database, and set permissions as follow:

```
mysql> create user 'radius'@'localhost' identified by 'setupRADIUS';
Query OK, 0 rows affected (0.00 sec)

mysql> create database radius;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on radius.* to 'radius'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
```
11) Then, we are going to create our database using phpmyadmin

We open a terminal and run the following commands:

```
cd /etc/freeradius/sql/mysql/
```

in order to go to the proper directory

```
gedit schema.sql
```

with this command the schema.sql file will open with the gedit text editor

![Command to open schema.sql](image1)

we select the radacct table like in the screenshot that follows and then right click copy’.

![Table structure for radacct](image2)

We open a browser and we point it to [http://localhost/phpmyadmin](http://localhost/phpmyadmin) we use the username and password we chose during the installation. On the left column we should see the radius database we created on step 10 and we click on that.
We select SQL like in the pic that follows, we paste the table from schema.sql file and hit go. We repeat this procedure for all the following tables in the schema.sql file (radcheck, radgroupcheck, radgroupreply, radpostauth, radreply, radusergroup).

Then we return to our terminal and run:

```bash
gedit nas.sql
```

in order to copy the nas table to our radius database as we did with the previous tables.
Finally our radius database should look like the picture that follows

12) Configure the file `/etc/freeradius/sites-available/default` so that the FreeRADIUS can startup, creates a default virtual host, and connects to MySQL Database. We follow the below steps:

Run the command:

```
sudo gedit /etc/freeradius/sites-enabled/default
```

In the file “default”, you check if the following settings are correct or not (the settings in the belowpictures are correct settings):

In the `authorize{}` module
In the accounting{} module

```
# Look in an SQL database. The schema of the database
# is meant to mirror the "users" file.
#
# See "Authorization Queries" in sql.conf
sql
#
# If you are using /etc/smbpasswd, and are also doing
# mschap authentication, the un-comment this line, and
# configure the 'etc_smbpasswd' module, above.
# etc_smbpasswd
```

(uncomment the sql setting)

In the session{} module

```
# sradutmp
#
# Return an address to the IP Pool when we see a stop record.
# main_tool
#
# Log traffic to an SQL database.
# 
# See "Accounting queries" in sql.conf
sql
#
# If you receive stop packets with zero session length,
# they will NOT be logged in the database. The SQL module
# will print a message (only in debugging mode), and will
```

In the session{} module

```
# Session database, used for checking Simultaneous-Use. Either the radutmp
# or rlm_sql Module can handle this.
# The rlm_sql module is *much* faster
session {
    radutmp
    
    # See "Simultaneous Use Checking Queries" in sql.conf
sql
}
```

# Post-Authentication
# Once we KNOW that the user has been authenticated, there are
In the post-auth{} module

We always save each file before we exit.

13) Now, we configure the “radiusd.conf”. This file contains general settings for the Radius Server. We follow the below steps:

```
sudo gedit /etc/freeradius/radiusd.conf
```

Then, we check if the settings in the file radiusd.conf are correct or not.

The port for radius server to listen for authentication request is 1812

Port for accounting is 1813

Some settings for logging username, password, etc
Then we make sure that freeradius will not read clients from the clients.conf file

And also that the sql.conf file will be included
14) Now, we configure the file “sql.conf”. This file contains information about how to connect to database, database tables that are used by the FreeRADIUS server, etc.

Run the command:

**sudo gedit /etc/freeradius/sql.conf**

Then, you should check the following settings:

Check the setting that allow FreeRADIUS to read Radius Clients from database.
15) We configure inner tunnel

Run the command

```bash
sudo gedit
```

we remove '#' before sql in the authorize{} module

![inner-tunnel configuration file](image)

16) Test if the Freeradius Server is working properly

Run the commands:

```bash
/etc/init.d/freeradius stop
```

```
* Stopping FreeRADIUS daemon freeradius
* /var/run/freeradius/freeradius.pid not found...                       [ OK ]
```

**freeradius -X** This command runs freeradius in debug mode and it is very helpful if we want to find out what went wrong.

If everything went well so far, the following picture is the expected response.
Configuring the Freeradius Server

We begin by creating our clients (our Access Points)
We open our browser and point it to http://localhost/phpmyadmin we select the radius
database we have already created and we click on the ‘nas’ table we select ‘Insert’
and we fill the blanks as in the following picture.

id : it will be filled automatically
nasname : the ip-address of our access point
shortname : the ssid of our access point
type : other
ports : 1812
secret : anything you want (we will use it when we will configure our access
point)
description : RADIUS Client
Then we will have to create our users. We select the radcheck table and we fill the blanks as in the following picture.

- **username**: the username of our user
- **attribute**: Cleartext-password
- **op**: := (this symbol means that any value that has already potentially been installed as Cleartext-password for our user will be overwritten by the new one. The simple equation (=) symbol wouldn’t overwrite any previous value.)
- **value**: our password

and click go to insert the new user.

As we can see here we have installed three users to our freeradius-mysql server.
Then we configure our groups of users. We select the radusergroup and by using the insert option we simply define which user belongs to what group.

As we can see we can create unlimited combinations of users and groups very easily.

We select the radreply table and then insert in order to configure our users.

- **username**: we select the user we want
- **attribute**: we select what attribute we want to change
- **op**: we select our operation
- **value**: we give a value to our attribute
In the following picture we can see what adjustments we made for the user Nick Niskopoulos. This is the default configuration for each user.

- All entries are processed in the order as they appear in the users file or in the radcheck table since we use MySQL database instead of the users file. If an entry matches the user name, radiusd will stop scanning the users file unless the attribute "Fall-Through = Yes" is set.
- Service-Type = Framed-User (with this rule only framed-user service types are allowed – no telnet, rlogin, tcp-clear–)
- Framed-Compression : = Van-jacobsen-TCP-IP . This framed compression attribute indicates a compression protocol to be used for the link. It may be used in Access-Accept packets. It may be used in an Access-Request packet as a hint to the server that the NAS would prefer to use that compression, but the server is not required to honor the hint. More than one compression protocol attribute may be sent. It is the responsibility of the NAS to apply the proper compression protocol to appropriate link traffic.
Following we select the radgroupcheck table and the ‘Insert’ tab and we set the authentication method for all users of the group ‘users’ to be EAP.

![Picture 25: configuring the authentication method](image)

Finally we select the radgroupreply table to configure the Framed-Protocol for all users of the group ‘users’ to be PPP. This rule allows PPP sessions (no SLIP, CSLIP, etc.).

![Picture 26: Configuring the Framed-Protocol the group users](image)

**Testing our configuration**

This is the basic configuration for the Freeradius Server that works with EAP authentication methods. In order to test our installation we can download the NTRadPing 1.5 RADIUS Test Utility from here: [http://www.novell.com/coolsolutions/tools/14377.html](http://www.novell.com/coolsolutions/tools/14377.html) and make the following adjustments.

We create a new client ‘NTradping Test’ in the NAS table of our database. (We set as nasname the ip of the operating system that the test utility will run)
We create a new user ‘ping’ with password 1234

We create a group ‘radping’ for our user ‘ping’.

We set the Authentication Type for our group ‘radping’ to PAP. This means that in order to authenticate the users, only username and password is required.
We make the same configuration that we did before only this time our settings are valid for all users of radping group.

Finally in radreply table we set ‘ping’ user to Fall-Through : = Yes
Every time we make any change to the freeradius configuration we run:

```
/etc/init.d/freeradius stop
```

and

```
freeradius -X
```

We run the NTRadPing Test Utility we have downloaded and make the following configuration:

- **RADIUS Server/port**: we set our radius server IP and the authentication port
- **RADIUS Secret Key**: The Key of NTradping Test client we created before
- **User-Name**: The new user we created
- **Password**: The users Password

And click Send. If everything went well we should see response: Access-Accept as in the following picture.

![Picture 33: The NTRadtest Utility](image)

### 802.1X with EAP-TLS Authentication

#### How to create the Certificates

On our Ubuntu system we open a new terminal and run the following commands:

```
sudo su
to run as administrator
```
**cd /etc/ssl** to change directory and go to ssl folder

**mkdir newcerts private users crl** to create these new folders in the ssl directory

**touch serial** to create a file named serial

**gedit serial** to open the serial file with the gedit text editor (write 01 save and close the file)

**touch ./index.txt** to create an empty txt file named index

**gedit openssl.cnf** to openssl.cnf file. Change

```
[ CA_default ]
dir = ./demoCA # Where everything is kept
to
dir = ./ # Where everything is kept
```

save and close the file.

**touch xpextensions** to create a file named xpextensions

**gedit xpextensions** open xpextensions and write

```
[xpclient_ext]
extendedKeyUsage=1.3.6.1.5.5.7.3.2

[xpserver_ext]
extendedKeyUsage=1.3.6.1.5.5.7.3.1
```

save and close the file. This is a workaround for a bug in Windows XP SP2 which could have negative impact on the TLS process if a third-party radius server is used like Freeradius.

Now we will use openssl to create our certificates

**Generating the ROOT-CA-Certificate**

```bash
openssl req -new -x509 -keyout private/cakey.pem -out cacert.pem -days 1095 -config openssl.cnf
```

```
Generating a 1024 bit RSA private key
........++++++
..................................................++++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase: the root ca password
```
Verifying - Enter PEM pass phrase: the root ca password
-----
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) [AU]:GR
State or Province Name (full name) [Some-State]:Athens
Locality Name (eg, city) []:Piraeus
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PA.PEI
Organizational Unit Name (eg, section) []:Digital Systems
Common Name (eg, YOUR name) []:NISKOPOULOS
Email Address []:radius@gmail.com

openssl pkcs12 -export -in cacert.pem -inkey private\cakey.pem -out caroot.p12 -cacerts -descert

Enter pass phrase for private\cakey.pem: the root ca password
Enter Export Password: caroot_p12_password
Verifying - Enter Export Password: caroot_p12_password
openssl pkcs12 -in caroot.p12 -out caroot.pem
Enter Import Password: caroot_p12_password
MAC verified OK
Enter PEM pass phrase: caroot_pem_password
Verifying - Enter PEM pass phrase: caroot_pem_password

openssl x509 -in cacert.pem -inform PEM -out cacert.der -outform DER
<There is no output for this command>

Generating the radius server request certificate

openssl req -nodes -new -x509 -keyout radius-req.pem -out radius-req.pem -days 730 -config openssl.cnf
Generating a 1024 bit RSA private key
................+++
...+++++
writing new private key to 'radius-req.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) [AU]: GR
State or Province Name (full name) [Some-State]: Athens
Locality Name (eg, city) []: Piraeus
Organization Name (eg, company) [Internet Widgits Pty Ltd]: PA.PEI.
Organizational Unit Name (eg, section) []: Digital Systems
Common Name (eg, YOUR name) []: Radius Server
Email Address []: niskopoulos@gmail.com

openssl x509 -x509toreq -in radius-req.pem -signkey radius-req.pem -out radius-tmp.pem
Getting request Private Key
Generating certificate request

Now we certify the request certificate

openssl ca -config openssl.cnf -policy policy_anything -out radiuscert.pem -extensions x50server_ext -extfile xpextensions -infiles radius-tmp.pem
Using configuration from openssl.cnf
Enter pass phrase for //private/cakey.pem: the root ca password
Check that the request matches the signature
Signature ok
Certificate Details:

Serial Number: 1 (0x1)
Validity
   Not Before: May 23 23:27:50 2012 GMT
   Not After : May 23 23:27:50 2013 GMT
Subject:
   countryName = GR
   stateOrProvinceName = Athens
   localityName = Piraeus
   organizationName = PA.PEI.
   organizationalUnitName = Digital Systems
   commonName = Radius Server
   emailAddress = niskopoulos@gmail.com

X509v3 extensions:
   X509v3 Extended Key Usage:
   TLS Web Server Authentication
Certificate is to be certified until May 23 23:27:50 2013 GMT (365 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n]: y
Write out database with 1 new entries
Data Base Updated
Creating client certificates (We have to repeat this step for every user)

```bash
openssl req -nodes -new -x509 -keyout nick.nisko-req.pem -out nick.nisko-req.pem -days 730 -config openssl.cnf
```

Generating a 1024 bit RSA private key
```
......................................++++++
........++++++
```

writing new private key to 'nick.nisko-req.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) [AU]:GR
State or Province Name (full name) [Some-State]:Athens
Locality Name (eg, city) []:Piraeus
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PA.PEI
Organizational Unit Name (eg, section) []:Digital Systems
Common Name (eg, YOUR name) []:NISKOPOULOS NIKOS
Email Address []:niskopoulos@gmail.com
```

```bash
openssl x509 -x509toreq -in nick.nisko-req.pem -signkey nick.nisko-req.pem -out nick.nisko-tmp.pem
```

Getting request Private Key
Generating certificate request

```bash
openssl ca -config openssl.cnf -policy policy_anything -out nick.nisko-cert.pem -extensions xpserver_ext -extfile xpextensions -infiles nick.nisko-tmp.pem
```

Using configuration from openssl.cnf
Enter pass phrase for .//private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
```
Serial Number: 2 (0x2)
Validity
    Not Before : May 23 23:50:21 2012 GMT
    Not After  : May 23 23:50:21 2013 GMT
Subject:
countryName                  = GR
stateOrProvinceName          = Athens
localityName                 = Piraeus
organizationName             = PA.PEI
organizationalUnitName       = Digital Systems
commonName                   = NISKOPOULOS NIKOS
eлектMailAddress             = niskopoulos@gmail.com
```
X509v3 extensions:
  X509v3 Extended Key Usage:
    TLS Web Server Authentication
Certificate is to be certified until May 23 23:50:21 2013 GMT (365 days)
Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated

openssl pkcs12 -export -in nick.nisko-cert.pem -out nick.nisko-cert.p12 -inkey nick.nisko-req.pem -desert
Enter Export Password: nick.nisko_password
Verifying - Enter Export Password: nick.nisko_password

openssl pkcs12 -nodes -in nick.nisko-cert.p12 -out nick.nisko-key.pem
Enter Import Password: nick.nisko_password
MAC verified OK

*Now we create a Diffi-Hellmann-Parameter (DH)*

openssl dhparam -out dh1024.pem 1024
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
                       +..........................     .....   ++*++*++*

Freeradius configuration for EAP-TLS

*Configuring the wireless router (Client)*

We are going to use a Pirelli wireless router but the basic configuration is approximately the same for all wireless access points.

- We open a browser and we login to our access-point. We select Wireless>Security and we set authentication type to 802.1X.
- We set the IP of our Freeradius Server, in our case 192.168.1.50 (we suggest that we set a static IP to the radius server)
- We set the Server-Port to 1812
- We use the Secret Key that we used when we configured the NAS table on our database, in our case setupRADIUS.
- We set the NAS-ID to other
- Finally we save our settings

![Picture 34: The Pirelli router setup web page]

### Configuring the Radius Server

We will have to copy `radius-req.pem`, `radiuscert.pem`, `cacert.pem`, `dh1024.pem` that we have created in the `/etc/ssl` directory into `/etc/freeradius/certs` directory. When we do that, we open a terminal:

```
cd /etc/freeradius/  # to change directory

gedit eap.conf  # to open eap.conf configuration file.
```

Make the following changes:

```
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 = tls

cisco_accounting_username_bug = yes

tls {

# These is used to simplify later configurations.
#
certdir = ${confdir}/certs
cadir = ${confdir}/certs

private_key_password = 1234
private_key_file = ${certdir}/radius-req.pem

# If Private key & Certificate are located in
# the same file, then private_key_file &
# certificate_file must contain the same file
# name.
#
# If CA_file (below) is not used, then the
# certificate_file below MUST include not
# only the server certificate, but ALSO all
# of the CA certificates used to sign the
# server certificate.
certificate_file = ${certdir}/radius-cert.pem

# Trusted Root CA list
#
# ALL of the CA's in this list will be trusted
# to issue client certificates for authentication.
#
# In general, you should use self-signed
# certificates for 802.1x (EAP) authentication.
# In that case, this CA file should contain
# *one* CA certificate.
#
# This parameter is used only for EAP-TLS,
# when you issue client certificates. If you do
# not use client certificates, and you do not want
# to permit EAP-TLS authentication, then delete
# this configuration item.
CA_file = ${cadir}/cacert.pem

# For DH cipher suites to work, you have to
# run OpenSSL to create the DH file first:
#
#   openssl dhparam -out certs/dh 1024
#
#   dh_file = ${certdir}/dh1024.pem
random_file = /dev/urandom

fragment_size = 1024
include_length = yes    remove #
check_cert_cn = %{User-Name}

save and close the window.

**gedit radiusd.conf** to open radiusd file.

prefix = /usr
eexec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbin = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
certdir = ${sysconfdir}/freeradius/certs
   add this line
What a client needs to authenticate with EAP-TLS method

Linux:
nick.nisko-cert.pem
nick.nisko-req.pem
cacert.pem

Windows:
nick.nisko-cert.p12
cacert.der

We are going to test to authenticate a user from win7 operating system. We copy these two files to our system nick.nisko-cert.p12, cacert.der and first we double click on the cacert file and click install certificate>Next>leave the default “Automatically select the certificate store based on the type of certificate” and Next>Finish.
Then we do the same procedure for the users’ certificate. Double click on nick.nisko-cert.p12>Next>Next>insert thenick.nisko_passwordand make sure the “Enable strong private key protection” is NOT checked>and Finish.
Now we are going to configure the wireless setting in Windows in order to authenticate the user by using the certificates. We click on our wireless connections > right click on our access point > properties > we change the Security type to WPA2-Enterprise Encryption Type to AES Network authentication method to Microsoft: smartcard or other certificate

Click on settings and check the Trusted Root Certification Authority that you have already inserted, click OK. Then click on Advanced settings and check the following configuration for 802.1X and 802.11 settings, then click OK.
Go to your wireless access point and click connect. After a few seconds a message like the one above will appear click on it and the Select Certificate Window will appear.

Select the certificate you have created and click OK.
We finally connect to our wireless network. The radius server will automatically authenticate you and the response ‘Access-Accept’ of the radius server in debug mode will be like in the following picture.
Radius Server Response

The complete response is approximately 15 pages here are the most interesting parts:

rad_recv: Access-Request packet from host 192.168.1.1 port 33568, id=3, length=165
   User-Name = "Nick Niskopoulos"
   NAS-IP-Address = 0.0.0.0
   Called-Station-Id = "00-1C-A2-B5-86-A1:ONTelecoms"
   Calling-Station-Id = "00-21-5D-4D-A3-50"
   NAS-Identifier = "other"
   NAS-Port = 29
   Service-Type = Framed-User
   Framed-MTU = 1400
   NAS-Port-Type = Wireless-802.11
   EAP-Message = 0x02030015014e69636b204e69736b6f706f756c6f73
   Message-Authenticator = 0xcdf863571f57a51d752574111efeb8d

...[eap] EAP packet type response id 3 length 21

...[sql] expand: SELECT id, username, attribute, value, op           FROM radcheck           WHERE username = '%{SQL-User-Name}'           ORDER BY id
     SELECT
     id, username, attribute, value, op           FROM radcheck           WHERE
     username = 'Nick Niskopoulos'           ORDER BY id

[sql] User found in radcheck table

...[sql] sql_set_user escaped user --> 'Nick Niskopoulos'
rlm_sql (sql): Reserving sql socket id: 4

Found Auth-Type = EAP
[eap] processing type tls
[eap] EAP Identity
[tls] Requiring client certificate
[tls] Initiate
[tls] Start returned 1
++[eap] returns handled

Sending Access-Challenge of id 3 to 192.168.1.1 port 33568
Service-Type := Framed-User
Framed-Compression := Van-Jacobson-TCP-IP
Framed-Protocol := PPP
EAP-Message = 0x010400060d20
Message-Authenticator = 0x00000000000000000000000000000000
State = 0x248c22ae24882fb57eb42a13aa8e525b

Finished request 4.
Going to the next request
Waking up in 4.9 seconds.

rad_recv: Access-Request packet from host 192.168.1.1 port 33569, id=4, length=267
User-Name = "Nick Niskopoulos"
NAS-IP-Address = 0.0.0.0
Called-Station-Id = "00-1C-A2-B5-86-A1:ONTelecoms"
Calling-Station-Id = "00-21-5D-4D-A3-50"
NAS-Identifier = "other"
NAS-Port = 29
Service-Type = Framed-User
Framed-MTU = 1400
NAS-Port-Type = Wireless-802.11
State = 0x248c22ae24882fb57eb42a13aa8e525b
EAP-Message
    0x020400690d80000000f160301005a0100005603014fbe272d545bf7b1d187
2459b09e357bba0eab025405d36258ef36c733ed016100001802f0035000500
0ac013c014c0099c0a03200380013000401000015ff01000100000a00060004
0170018000b00020100

Sending Access-Challenge of id 4 to 192.168.1.1 port 33569
Service-Type := Framed-User
Framed-Compression := Van-Jacobson-TCP-IP
Framed-Protocol := PPP
EAP-Message
    0x010504000dc0000006f716030100310200002d03014fbe272d8c32e3499c8b
d294cc859df30e3c5bf406dd9db1645582c95adac0002f000005ff01000100
016030105e70b0005e30005e00002cc308202c83082023a0030201020201013
00d06092a8e6488ef70d01010505030819d310b300906035504061302475231
0f300d06035504080c0641748656e733110f300d0603550407c0641746856e7331
133011060355040a0c0a556e2e5069726165757331183016060355040b0
c0f449676974616c5f3797374656d733115301306035504030c05c26164697
5735f4e9736b6f3126320406092a86488f70d0109011676e9736b6f70
EAP-Message
    0x6f756c6f7340686f746d61696c2e636f6d301e170d31323035313031353035
0315a170d31333035313031353035303135a30819c310b300906035504061302
4752310f300d06035504080c0641748656e733110f300d0603550407c064174
68656e7331133011060355040a0c0a556e2e506972616575733118301606035

rad_recv: Access-Request packet from host 192.168.1.1 port 33570, id=5, length=168
User-Name = "Nick Niskopoulos"
NAS-IP-Address = 0.0.0.0
Called-Station-Id = "00-1C-A2-B5-86-A1:ONTelecoms"
Calling-Station-Id = "00-21-5D-4D-A3-50"
NAS-Identifier = "other"
NAS-Port = 29
Service-Type = Framed-User
Framed-MTU = 1400
NAS-Port-Type = Wireless-802.11
State = 0x248c22ae25892f57eb42a13aa8e525b
Message-Authenticator = 0x00000000000000000000000000000000
State = 0x248c22ae25892f57eb42a13aa8e525b

Finished request 5.
Going to the next request
Waking up in 4.9 seconds.
Sending **Access-Accept** of id 7 to 192.168.1.1 port 33572

Service-Type := Framed-User
Framed-Compression := Van-Jacobson-TCP-IP
Framed-Protocol := PPP
MS-MPPE-Recv-Key = 0x3153473966dbf4e13954ae29742a624fe1749e79f398d34b7acfd8d979e4974
c
MS-MPPE-Send-Key = 0xe7142d8aa5465f9c7c9233f5e2cb1b43a9c89d655f2c02621326e19c0dbb6fc
EAP-Message = 0x03070004
Message-Authenticator = 0x00000000000000000000000000000000
User-Name = "Nick Niskopoulos"

Finished request 8.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 8 ID 7 with timestamp +2977
Ready to process requests.

As we can see everything works properly, the sql database found both the client and the user and the tls authentication method was completed successfully.
Freeradius with LDAP on Linux Ubuntu 11.10

**Freeradius with LDAP**

A Radius Server, is a daemon for unix operating systems which allows one to set up a radius protocol server, which is usually used for authentication and accounting of dial-up users. To use server, you also need a correctly setup client which will talk to it, usually a terminal server or a PC with appropriate which emulates it (PortSlave, radiusclient etc).

Radius has its own database of users since this information is already contained in LDAP it is far more convenient to use its advantages. There are several freeware Radius servers, the one that has good support for LDAP and will be used in this implementation is the FreeRadius server.

LDAP stands for Lightweight Directory Access Protocol. LDAP is a protocol which provides access to a compliant directory via TCP/IP. The strengths of LDAP-compliant directories include speed, simplicity, and the ability to be replicated and distributed across several servers. A LDAP directory can be used to store a great deal of information: from user login credentials to company telephone directories.

LDAP was created as a less complicated implementation of the Directory Access Protocol (DAP), and is based on the OSI X.500 standard. These standards establish directories as being hierarchical—representing the structure of an organization. There are many directories that support the LDAP protocol, each with their own benefits and drawbacks. Examples include openLDAP, the open-source implementation that ships with SLES and eDirectory, Novell's flagship identity management product.

Why LDAP instead of MySQL? The largest general difference between directories and databases is complexity. Databases are capable of storing almost any arbitrary set of information and can be greatly customized for a specific purpose. They also provide a complex query interface, allowing for flexible searches returning customized results. Directories, on the other hand, tend to have very specific implementations that follow a strict pattern or schema. This allows them to be extremely fast, and allows for easy organization and comprehension of the data they store.
Installing Freeradius with LDAP

1. We start by running the commands as admin

   `sudo su`

2. We install freeradius

   `apt-get install freeradius`

3. And install freeradius-ldap

   `apt-get install freeradius-ldap`

4. We install the openldap server and the necessary utilities

   `apt-get install slapd ldap-utils`

5. We reconfigure the ldap server to make some changes

   `dpkg-reconfigure slapd`

   ```
   [ 
   Omit OpenLDAP…> NO
   DNS domain name..> example.com
   Organization name..> Nick Nisko (anything you want)
   Password..> 1234
   Confirm..>1234
   Database backend…> BDB
   NO
   YES
   NO
   ]
   ```
Now we start the freeradius configuration

1) We configure the client file of freeradius as below

```
  gedit /etc/freeradius/clients.conf
[
  //at the end of the file we add the clients that we want, these clients //that I have added are referred to my access point and the NTRadping //client that I use from my windows system to test the radius server
  Clinet 192.168.88.1 {
    Secret = 1234
    Shortname = radping
  }
  Clinet 192.168.1.1 {
    Secret = setupRADIUS
    Shortname = ONTelecoms
  }
]
```

2) We configure the ldap file

```
  gedit /etc/freeradius/modules/ ldap
[
]
```

3) Then we configure the default file

```
  gedit /etc/freeradius/sites-available/default
[
  authorize {
```
# Read the 'users' file
# files

ldap

authenticate {

Auth-Type LDAP {
    ldap
}

}

4) And also in the sites-available directory we configure the inner-tunnel file

gedit inner-tunnel

{
authorize
    # files

    ldap

    Auth-Type LDAP{

}
5) Final at the freeradius configuration we make the following changes at the radius.conf file

```bash
gedit /etc/freeradius/radiusd.conf
```

```
$Include clients
```

### Installing phpladapadmin

1. Now we should install phpladapadmin in order to easily access and configure our ldap database.

   **apt-get install phpladapadmin**

   //This is a web based tool which helps us to manage our LDAP database. For the
   //former approach, just edit /etc/phpladapadmin/config.php. You'll probably need to
   //change the following lines (the most important is to add your own ip address of the
   //ldap server instead of the highlighted one):

   ```php
   $ldapservers->SetValue($i, 'server','name','My LDAP Server');  // The name to display
   $ldapservers->SetValue($i, 'server','host','192.168.1.50');  // Address of the LDAP server
   $ldapservers->SetValue($i, 'server','port','389');  // Port number
   $ldapservers->SetValue($i, 'base','array','dc=example,dc=com');  // Base dn
   $ldapservers->SetValue($i, 'login','string','uid=<username>,ou=People,dc=example,dc=com');
   ```

2. We make the following changes in the ldap server’s ldap.conf file

   ```bash
   gedit /etc/ldap/ldap.conf
   ```
{ 
BASE  dc=example, dc= com
URI  ldap://192.168.1.50
}

( 
Go and check on a browser 192.168.1.50/phpldapadmin
If error ( Your PHP memory limit is low-currently 16M):
gedit /etc/php5/apache2/php.ini
{
memory_limit = 32M
}
Invoke-rc.d apache2 restart
)

3. We want to restart our radius server so we open a new terminal and run the commands

/etc/init.d/freeradius stop
freeradius -X

(Should you have any problem with permissions "Opening file : ...... Permission denied" try
sudo /etc/init.d/freeradius stop
sudo freeradius -X  )

NTRadping Test Utility
We will use NTRadping to test if all the configuration is correct

We use NTRadping tool on our windows OS and send an authentication request to test if our radius server works (instructions regarding the NTRadPing tool and its installation can be found in the paragraph). As we can see our server works properly.

We use the ip address of our radius server and the authentication port 1812, we use the Radius Secret key that we have entered for our windows (radping) client and for User-Name user and password 1234 we receive a response: Access Reject since we haven’t yet created such a user in our LDAP database, as we can see in the following picture.

![Picture 44:Testing the curent configuration with NTRadPing Test utility](image)

Configuring the LDAP Directory

Now we can open a browser and go to localhost/phpldapadmin/ in order to manage our ldap database
**How to create user user1**

We will create user1 and we will configure our LDAP database in order to make a successful authentication.

Open a browser and locate it to [localhost/phpldapadmin](http://localhost/phpldapadmin)

Click on login and use the password you used during the ldap server installation

![Picture 45: Login to LDAP Directory](image)

Click on ‘Create new entry’ and then select Generic: Organizational Unit

![Picture 46: Creating new entry](image)

Name the new organizational unit people and click ‘Create Object’

![Picture 47: Creating a new object](image)

Click on ‘Commit’
Then we create by following the same procedure another organizational unit called ‘groups’.

We click on ou=groups and select ‘create a child entry’

Then we select ‘generic : Posix Group’

We name our Posix Group adminuser
Then we will create a user under the people organizational unit. Click on people and then click on ‘Create a new child entry’

We select to create a new user account

We create our user by filling the above form as we can see here and click ‘Create Object’
Simply if we agree with our user that we just created we press ‘Commit’.

Using NTRadping to test if all the configuration is correct

We use NTRadping tool on our windows OS and send an authentication request to test once more if our radius server works, only this time we expect to be successfully authenticated.

We use the ip address of our radius server and the authentication port 1812, we use the Radius Secret key we have entered for our windows (radping) client and for User-Name user1 and password 1234.

The proof of concept, as we can see in the screenshot above, is the response: Access Accept.
And on our terminal on our Ubuntu OS we will see the above response of our freeradius-ldap server (running in debug mode).

```
rad_recv: Access-Request packet from host 192.168.1.2 port 63051, id=5, length=45
User-Name = "user1"
User-Password = "1234"

# Executing section authorize from file /etc/freeradius/sites-enabled/default
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
++[digest] returns noop
[suffix] No '@' in User-Name = "user1", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
```
++[eap] returns noop

[ldap] performing user authorization for user1

[ldap] expand: %{Stripped-User-Name} ->

[ldap] ... expanding second conditional

[ldap] expand: %{User-Name} -> user1

[ldap] expand: (uid=%{Stripped-User-Name}:-%{User-Name}) -> (uid=user1)

[ldap] expand: dc=example, dc=com -> dc=example, dc=com

[ldap] ldap_get_conn: Checking Id: 0

[ldap] ldap_get_conn: Got Id: 0

[ldap] attempting LDAP reconnection

[ldap] (re)connect to 192.168.1.50:389, authentication 0

[ldap] bind as cn=admin, dc=example, dc=com/1234 to 192.168.1.50:389

[ldap] waiting for bind result ...

[ldap] Bind was successful

[ldap] performing search in dc=example, dc=com, with filter (uid=user1)

[ldap] No default NMAS login sequence

[ldap] looking for check items in directory...

[ldap] userPassword -> Password-With-Header == "{MD5}gdyb21LQTclANtvYMT7QVQ=="

[ldap] looking for reply items in directory...

[ldap] Setting Auth-Type = LDAP

[ldap] user user1 authorized to use remote access

[ldap] ldap_release_conn: Release Id: 0

++[ldap] returns ok

++[expiration] returns noop

++[logintime] returns noop

[pap] Normalizing MD5-Password from base64 encoding
[pap] WARNING: Auth-Type already set. Not setting to PAP
++[pap] returns noop

Found Auth-Type = LDAP

# Executing group from file /etc/freeradius/sites-enabled/default
+- entering group LDAP {...}

[ldap] login attempt by "user1" with password "1234"

[ldap] user DN: cn=user1,ou=nikos,dc=example,dc=com

[ldap] (re)connect to 192.168.1.50:389, authentication 1

[ldap] bind as cn=user1,ou=nikos,dc=example,dc=com/1234 to 192.168.1.50:389

[ldap] waiting for bind result ...

[ldap] **Bind was successful**

[ldap] user user1 authenticated succesfully

++[ldap] returns ok

Login OK: [user1] (from client radping port 0)

# Executing section post-auth from file /etc/freeradius/sites-enabled/default

+- entering group post-auth {...}

++[exec] returns noop

**Sending Access-Accept of id 5 to 192.168.1.2 port 63051**

Finished request 0.

Going to the next request

Waking up in 4.9 seconds.

Cleaning up request 0 ID 5 with timestamp +11

Ready to process requests.
Creating Users by using JXplorer tool.

If we don’t want to use phpldapadmin to configure our ldap database there is another tool called JXplorer which can be easily downloaded and installed by using a package manager or the Dash Home tool on our Ubuntu 11.10.

Once you download this application and load it you should click on the upper left symbol of bond as you can see above.

Then we should fill which ldap directory we want to connect to and we will select the database we created during the installation and configured before with phpldapadmin. We insert the following info to the new window that will come up as in the screenshot that follows. The password is the same with the one used before (1234).
When our database is loaded we can see on the list the organizational users we created with phpldapadmin (this is a proof that the database functions properly). Then we right click on example and select New.

![Creating new organizational unit](image)

Now we can select what object we want to insert. We will create an organizational unit and name it new_org_unit (ou=new_org_unit). The parent is already set to be the ‘example.com’. From the available classes we will select ‘organizationalUnit’ and we will ‘Add’ it to our ‘Selected Classes’ (if any unwanted Classes are preselected we can easily select and remove them) as we can see in the screenshot.
If we agree with the organizational unit that we just created, we simply press ‘Submit’.

Then as we did before in order to create the organizational unit, we right click on ‘new_org_unit’ and select ‘New’. We fill the ‘Enter RDN’ with name of the user we
want to create ‘cn=user2’ and we select the classes inetOrgPerson, Person, posixAccount and click OK.

![Set Entry Object Classes](image)

**Picture 62: Creating the new user**

We fill the gidNumber, homeDirectory, sn, uid and uidNumber attributes as in the following picture and then Submit and change to the ‘HTML View’ tab.

![User Attributes](image)

**Picture 63: Creating the new user**

We fill the user password section with the desired Password (in this case 4321) click ‘Submit’ and Return to the ‘Table Editor’ tab.
Since we are back to Table Editor tab click on (no string data) in the userPassword Attribute. Fill in the password and the encryption algorithm we desire and then ‘OK’.
Now that we have created another user, we should check if we can be authenticated to the freeradius-ldap server with these new credentials. We open the NTRadping Test Utility on our Windows OS and as we did before we set the ip of the Radius server and the port 1812 (which is the default port for authentication), the radius server key that we have used during the installation for our windows client, the username of the user that we have just created and his password (4321) click next and the proof of concept reveals, ‘response: Access Accept’. We can see the response in the following screenshot.
Our freeradius-ldap server running in debug mode has the following output:

*Ready to process requests.*

*rad_recv: Access-Request packet from host 192.168.1.2 port 63121, id=1, length=45*

*User-Name = "user2"

*User-Password = "4321"

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] No '@' in User-Name = "user2", looking up realm NULL

[suffix] No such realm "NULL"

++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop

[ldap] performing user authorization for user2
[ldap] expand: %{Stripped-User-Name} ->
[ldap] ... expanding second conditional
[ldap] expand: %{User-Name} -> user2
[ldap] expand: (uid=%{Stripped-User-Name}:-%{User-Name}) -> (uid=user2)
[ldap] expand: dc=example, dc=com -> dc=example, dc=com

[ldap] ldap_get_conn: Checking Id: 0
[ldap] ldap_get_conn: Got Id: 0
[ldap] attempting LDAP reconnection
[ldap] (re)connect to 192.168.1.50:389, authentication 0
[ldap] bind as cn=admin, dc=example, dc=com/1234 to 192.168.1.50:389
[ldap] waiting for bind result ... 
[ldap] Bind was successful
[ldap] performing search in dc=example, dc=com, with filter (uid=user2)
[ldap] No default NMAS login sequence
[ldap] looking for check items in directory...

[ldap] userPassword -> Password-With-Header == 
"{MD5}2TWVeGDh5O4vynnZESFQ=="
[ldap] looking for reply items in directory...

[ldap] Setting Auth-Type = LDAP
[ldap] user user2 authorized to use remote access

[ldap] ldap_release_conn: Release Id: 0
++[ldap] returns ok
++[expiration] returns noop
++[logintime] returns noop
Normalizing MD5-Password from base64 encoding

WARNING: Auth-Type already set. Not setting to PAP
++[pap] returns noop

Found Auth-Type = LDAP

# Executing group from file /etc/freeradius/sites-enabled/default
+- entering group LDAP {...}

[ldap] login attempt by "user2" with password "4321"

[ldap] user DN: cn=user2,ou=new_org_unit,dc=example,dc=com

[ldap] (re)connect to 192.168.1.50:389, authentication 1

[ldap] bind as cn=user2,ou=new_org_unit,dc=example,dc=com/4321 to 192.168.1.50:389

[ldap] waiting for bind result ...

[ldap] Bind was successful

[ldap] user user2 authenticated succesfully
++[ldap] returns ok

Login OK: [user2] (from client radping port 0)

# Executing section post-auth from file /etc/freeradius/sites-enabled/default
+- entering group post-auth {...}

++[exec] returns noop

Sending Access-Accept of id 1 to 192.168.1.2 port 63121

Finished request 0.

Going to the next request

Waking up in 4.9 seconds.

Cleaning up request 0 ID 1 with timestamp +9

Ready to process requests.
PAM/USB Module

Introduction

PAM-USB module is an authentication method which uses an external USB flash drive as a token to authenticate a user to a service or to the operating system that he uses. Here, an operating system authentication will be implemented were the OS will be Ubuntu 11.10 and the pamusb tool is going to be used.

Installing PAM/USB Module

1) Install the pam usb

We download the zip file (aluzzardi-pam_usb-0.5.0-23-g8d630b9.zip) to our desktop from http://pamusb.org/ and we extract the file into our Desktop (or anywhere we like since we remember were we saved it)

```
cd /home/youruser/Desktop
unzip aluzzardi-pam_usb-0.5.0-23-g8d630b9.zip
```

```
cd aluzzardi-pam_usb-8d630b9
```

Before we install the source code we have to install all of the above in order to avoid some problems they came up during the installation

```
sudo apt-get install udisks
```

```
sudo apt-get install pmount
```

```
sudo apt-get install libxml2
```

```
sudo apt-get install libxml2-dev
```

```
sudo apt-get install libdbus-1-dev
```

```
sudo apt-get install libpam0g-dev
```

```
2) We run

```bash
cd /home/youruser/Desktop/ aluzzardi-pam_usb-8d630b9 //or wherever you installed //the -aluzzardi-
pam_usb- //file you downloaded on //step 1
make
make install  // if everything went well the installation should finish without any problems
```

3) We insert our usb flash drive into a usb port of our computer

Important! If we use a virtual machine we have to make sure that our device is installed to the guest operating system and not to host. It might need to restart your OS if the usb device does not appear to your guest OS.

4) We select the usb flash drive that we will use

```bash
pamusb-conf --add-device=TRAVELDRIVE  //use the exact name of the device //we will use
```

![Picture 67: TRAVELDRIVE usb flash drive is inserted to our PC](image)

niko@ubuntu:~$ pamusb-conf --add-device=TRAVELDRIVE
Please select the device you wish to add.
* Using "Memorex TD Classic 083B (Memorex_TD_Classic_083B_076A1892905E-0:0)" (only option)

Which volume would you like to use for storing data ?
* Using "/dev/sdb1 (UUID: E60D-E84F)" (only option)

Name : TRAVELDRIVE
Vendor : Memorex
Model : TD Classic 083B
Serial : Memorex TD Classic_083B_076A1892905E-0:0
UUID : E60D-E84F

Save to /etc/pamusb.conf ?
[Y/n] Y
5) We add a user

```
pamusb-conf --add-user=nik0z //use the exact name of the user we want to
//authenticate the system is case sensitive
//Nik0z≠nik0z
```

Picture 68: OS user nik0z

6) We can add another user if we want to the same device

```
pamusb-conf --add-user= user1
```

As we can see in the above screenshot after we run the command we can select in
witch device we want to add our user (I have already installed two usb devices
TRAVELDRIVE and PAMUSB-1) here we select 0 the flash drive PAMUSB-1 and
then we select Y in order to write the new information into pamusb.conf file.

![Adding a user to device PAMUSB-1](image)

7) Now modify the common-auth file in order to start using the usb flash to
authenticate ourselves while entering the system.

```
getic /etc/pam.d/common-auth
```

8) Then we write inside

```
auth sufficient pam_usb.so
```

as in the following picture
9) **Check if the usb works**

We open a new terminal and simply run the command

```
sudo su
```

We can disable this feature from the `/etc/pamusb.conf` file in the `services` section by adding the following:

```
<service id="su">
    <!-- Disable pam_usb for "su" ("su" will ask for a password as usual) -->
    <option name="enable">false</option>
</service>
```

Or alternatively we run

```
pamusb-check nik0z
```

Instead of `nik0z` we use whatever username we have created.
10) We remove our usb flash drive and we switch user account.

As we can see a password is needed to authenticate user nik0z

We insert our usb flash drive that we have used to install user nik0z and wait a couple of seconds.

As we can see it no longer needs password to authenticate user nik0z, we simply press login.
Success!

Two-Factor Authentication

If we want to use two-way authentication (Password and USB Flash Drive) we have to make the following change.

1) We simply change the common-auth file

```bash
gedit /etc/pam.d/common-auth
```

2) We write inside

```bash
authentication required pam_usb.so
```
Instead of `authentication sufficient pam_usb.so`.

This means that both password and usb flash drive will be needed in order to authenticate a user (Two-Factor authentication)

We unplug our flash drive and change user account once again. As we can see in the following picture despite the fact that the correct password is inserted the authentication is not completed.
3) We insert our USB drive, wait a couple of seconds, and type again the correct password.

Then this time we will enter our OS.

Success!
LinOTP tool

Introduction

A one-time password (OTP) is a password that is valid for only one login session or transaction. OTPs avoid a number of shortcomings that are associated with traditional (static) passwords. The most important shortcoming that is addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attacks. This means that, if a potential intruder manages to record an OTP that was already used to log into a service or to conduct a transaction, he or she will not be able to abuse it since it will be no longer valid. On the downside, OTPs are difficult for human beings to memorize. Therefore they require additional technology in order to work. LinOTP is an open solution for strong two-factor authentication with One Time Passwords.

This new LinOTP 2 core modules and basic necessary components are licensed under the AGPLv3, so that you are able to have a complete working open source solution. But LinOTP2 is also open as far as its modular architecture is concerned. LinOTP 2 aims to not bind you to any decision of the authentication protocol or it does not dictate you where your user information should be stored. This is achieved by its new, totally modular architecture.

LinOTP also provides a modular architecture to calculate OTP values. Thus many different OTP algorithms can be supported by LinOTP. LinOTP is originally based on the HMAC-OTP algorithm. This open algorithm is defined in RFC4226. The HMAC-OTP algorithm calculates an OTP value on the basis of a secret key that is shared between the LinOTP server and the user. This secret key is not put plainly in the users hand but is forged into a piece of hardware device - the OTP token. LinOTP supports several different hardware and software OTP tokens.

Installing the linOTP2 tool

Because LinOTP2 doesn’t support the latest Ubuntu releases (11.04 and later) some problems might occur during the installation. We open a new terminal and type:

1) In order to get administrative permissions

```bash
sudo su
```

2) We update our repositories

```bash
apt-get update
```
apt-get upgrade

3) and we install all the following prerequisites in order to avoid as many installation bugs as possible.


(make sure you remember the mysql root password because we are going to need it later)

Mysql-server installation is recommended to achieve a more stable linotp installation. Using apache2 you get authentication and due to threading it is not possible to use SQLite with LinOTP running under apache2

4) Download the following packages:

   linotp_2.2_all.deb
   linotpuseridresolver_2.2_all.deb
   libpam-linotp_2.2_i386.deb
   linotpadminclientce_2.2-pre2_all.deb

   from here: http://www.linotp.org/index.php/download

5) Install the packages by issuing the command:

   dpkg -i linotp_2.2_all.deb linotpuseridresolver_2.2_all.deb libpam- linotp_2.2_i386.deb linotpadminclientce_2.2_all.deb

   (alternately you can choose to do it by using the gui of Ubuntu. You should go to the folder were these files were downloaded and >right click on linotpuseridresolver_2.2_all.deb >Open with GDebi Package Installer>Install package

   Do the same for the linotpadminclientce_2.2_all.deb and linotp_2.2_all.deb in that sequence!
No matter which method you will use to install the packages, when installing linotp_2.2_all.deb you will be asked several questions, make the following configuration.

Do you want to run LinOTP 2 via Apache2?
-> Yes
Enter admin password for the new LinOTP admin account.
-> think of one
Do you want to create a self-signed certificate?
-> Yes
What SQL database do you want to use for the token database?
-> Mysql
database hostname:
-> localhost

Name of Token database
-> LinOTP2

database user
-> linotp2

database users password:
-> think of one

Enter my sql root password
Do you want to create the tables
-> yes

(Should any problem appears, run dpkg in debug mode and try to locate and fix the problem)

dpkg --debug=3773 -i linotp_2.2_all.deb linotputeridresolver_2.2_all.deb libpam-
linotp_2.2_i386.deb linotpadminclientce_2.2_all.deb)

Then the encryption key /etc/linotp2/encKey will be created.

Restart the apache:

/etc/init.d/apache2 reload

You will be able to see the management interface by pointing your browser to:

https://localhost/manage/index

Once you do that you will have to accept the certificate that was just created

We click on ‘I understand the risks’

And ‘Add Exception’ as we can see in the following screenshot.
Then we simply ‘Confirm the Security Exception’.

For username we type ‘admin’ and for password the password we used during the linopt2 installation.
If you get an "Internal Server error", assure, that the complete directory /etc/linotp2 is owned by the user linotp

If everything goes fine the linotp2 web user interface will appear.

![Picture 84: LinOTP2 WEB User Interface](image)

**Configuration**

You may use the management web user interface or the command line client to setup your LinOTP server. You need to configure a UserIdResolver (a source were usernames/id’s will be extracted from your system) and a default realm. You may do this by issuing the following commands:

```
linotpadmin.py --url=https://localhost --admin=admin --command=setresolver --resolver=defaultPW --rtype=FILE --rf_file=/etc/passwd
```

That reads the users from the /etc/passwd file. You will get a JSON feedback like this:
Now you need to add this resolver to your default realm:

```
linotadmin.py --url=https://localhost --admin=admin --command=setrealm --realm=defRealm --resolver=useridresolver.PasswdIdResolver.IdResolver.defaultPW
```

You can either check it in the web UI:

```
linotadmin.py --url=https://localhost --admin=admin --command=setdefaultrealm --realm=defRealm
```

or issue the command
linotadmin.py --url=https://localhost --admin=admin --command=listuser

Now we are ready to assign tokens to our users.

**Enroll OTP Token**

You may want to enroll an OTP. In this example we enroll a motp token on our mobile phone (HTC Desire S android-2.3.5). This token will be generated to our phone every time we want to authenticate to the system. We used the app **mOTP token** which can be found downloaded and installed to your phone from the android market ‘Google Play’. We create our user by filling the name the pin we want (the pin will be inserted every time we would want to generate a new One Time Password) and the secret value of our choice (16 hex digits) like in the following screenshots.

This is the main page where we create new Profile.

![MOTP Profile](image)

We create the username, we add the pin we want (which is going to be asked every time we wish to generate a new OTP in our phone) and finally the secret value-seed which is going to be used in combination with our pin to create the One Time Password. Then we simply save our settings.
We then we will face the following tab were the pin that we have chosen will be asked and then we just click generate.

Our OTP is the six digits 1b3c07 as we can see in the following screenshot and it is valid for a finite time (in this case for the next 163 seconds).
If we don’t want to use a mobile device and we just want to test if our configuration works properly we can use the following url: http://motp.sourceforge.net/motp_js.html, were we can insert the pin and the secret we want to use and generate the One-Time-Password(OTP) as we can see in the following screenshot as we did with the phone application.

![MOTP passcode generator](image1)

Picture 91: MOTP passcode generator

Then we go back to our computer and create our user to our linotp server. This can be done either by using the web user interface where we select the user we want and click ‘enroll’.

![Create token to our server](image2)

Picture 92: Create a token to our Server

Then we select

- **Token Type**: mOTP
- **Init secret**: the same 16 hex digits we used on our phone application
- **mOTP**: 1234 (the same we will every time to out phone)

and click ‘Enroll’.
As we can see our first token has been created.

Or alternatively with the use the following command in terminal to avoid all the web user interface procedure.

```
linotpadmin.py --url=https://localhost --admin=admin --command=inittoken --user=nik0z (whatever user you like to authenticate to your Ubuntu OS) --type=motp --otpkey=3435febb9a5c44a6 (the 16 hex digits we used in our mobile device) --serial=1 --otppin=1234 (whatever pin we used in our mobile device)
```

This means, that you generated a mOTP token that gets assigned to the user "cornelius". The otpkey is the init-secret that was displayed on your phone. You may choose a serial (serial number) as you like to. The otppin is the PIN you will enter on the phone to generate an otp value.
TWO-factor Authentication

Optionally we may set an OTP PIN, which is a fixed-static password part that you need to enter in front (Depending on the parameter PrependPIN) of the OTP value.

By using this option we actually enable a two factor authentication procedure (something we know: static password and something we create in real time with our OTP generator: one time password from our phone)

With the following command our static password from now on will be 123.

`linotpadmin.py --url=https://localhost --admin=admin --command=set --pin=123 --serial=1`

Now we can test the token that we created. Insert to your browser the url: `https://localhost/auth/index` and you will enter the following page.

Enter your username and as password the OTP PIN we just created (in our case the simple pin 123) and then the 6 digit OTP value you generated with your phone. We don’t have to delay to enter the OTP because the OTP value that is generated to our phone is based to epoch (time in seconds since 1/1/1970), thus it is synchronized to our computer, consequently the OTP is valid for a few seconds after and after some time it will no longer be valid so we should generate a new one. If we fail to authenticate, we might mistyped the secret value or most likely the time of our phone is out of sync.

If everything goes well we should see a message like this to appear.
As we can see by the dots in the password field we used our pin (3 digits) + our OTP value (6 digits).

**Authenticate to our Operating System Desktop**

Now we will setup PAM to enable the ability to authenticate to our desktop via OTP. The security module /lib/security/pam_linotp.so is already installed to our system. We may now setup otp authentication. We choose to do it modular:

```bash
cp /etc/pam.d/common-auth /etc/pam.d/common-linotp
```

Just for safety we won’t use the common-auth file but instead we will copy all its contents (backup) to a new file called common-linotp and we will make there all the necessary changes.

Run the command

```bash
gedit /etc/pam.d/common-linotp
```

and change the line:

```bash
auth [success=1 default=ignore] pam_unix.so nullok_secure
```

to

```bash
auth [success=1 default=ignore] pam_linotp.so debug nosslhostnameverify
nossclcertverify url=https://localhost/validate/simplecheck realm=defrealm
resConf=defaultPW
```

Please note, that when using "debug" option, many information - also the Password - will be written to the auth.log. Now we may use the common-linotp in any PAM configuration instead of common-auth where we like to, here we will change the following lines in /etc/pam.d/gdm in order to authenticate to our system using OTP authentication.

Run the command

```bash
gedit /etc/pam.d/gdm
```

and change

```bash
@include common-auth
```

to

```bash
@include common-linotp
```
Now you will need to authenticate with OTP to our Gnome desktop! Note that we will also need to change /etc/pam.d/gnome-screensaver if we also want to unlock the desktop using OTP: You will not be asked for "Password" anymore but for "Your OTP", as we can see above.

And for the proof of concept we login to our desktop

If you fail to authenticate you may also take a look into /var/log/auth.log
**Triple-factor authentication**

Now we can implement a triple factor authentication by combining the linOTP2 tool with the PAM-USB module (Something we know: our pin, something we randomly create: OneTimePassword from our phone generator and finally something we possess: the USB-flash drive). As we did in the previous PAM-USB module section we download and install PAM-USB module from pamusb.org

Or if we are using Ubuntu 10.04 as we do here we simply open a terminal and run the command

`sudo apt-get install libpam-usb pamusb-tools`

and then the procedure is the same as follows:

in order to add a USB device

`pamusb-conf --add-device=TRAVELDRIVE`

to add a user we run the command

`pamusb-conf --add-user=nik0z`

and then we modify the common-auth file of our system. In this case we will modify the common-linotp file since we have replaced the common-auth file with the common-linop during this current configuration a few steps back. So we run the command

`gedit /etc/pam.d/common-linotp`

and we add the first line as follows

`auth required pam_usb.so`
We remove the USB flash drive from our computer and we try to login. As we can see the authentication fails (Permission Denied).

Then we insert our USB flash drive and we type again our password and the OTP from our mobile device. The proof of concept, we login to our Desktop.
Success!
References

Freeradius MySQL

- http://wiki.freeradius.org
- http://www.youtube.com/watch?v=gSCgESg8UvY
- https://help.ubuntu.com/community/UsingTheTerminal
- http://linuxtechtutorials.blogspot.com/
- http://www.openssl.org/support/faq.html
- http://www.dartmouth.edu/~pkilab/pages/EAP-TLSwFreeRadius.html
- http://www.privacywonk.net/2010/10/security-how-to-wpa2-enterprise-on-your-home-network.php
- http://www.youtube.com/watch?v=fVRDbAxKcqw&feature=related

Freeradius LDAP

- http://wiki.freeradius.org/
- http://wiki.freeradius.org/LDAP
- http://linux-ubuntu1104.blogspot.com/
- http://serverfault.com/questions/47815/setting-up-radius-ldap-for-wpa2-on-ubuntu
- http://www.youtube.com/watch?v=gSCgESg8UvY&feature=relmfu
- http://www.youtube.com/watch?v=Mw_1rGX635Y&feature=channel&list=UL
- http://www.youtube.com/watch?v=DM_UQVVVtoY

**PAM/USB Module**

- http://pamusb.org/
- http://csc560.wikispaces.com/PAM+USB
- http://www.youtube.com/watch?v=RDkK1quOfuE&feature=related
- http://www.youtube.com/watch?v=CCfwegqZlsoA
- https://github.com/aluzzardi/pam_usb/wiki/Install
- https://github.com/albuzzardi/pam_usb/wiki/Configuration
- https://github.com/aluzzardi/pam_usb/wiki/Getting-Started

**LinOTP**

- http://motp.sourceforge.net/
- http://motp.sourceforge.net/motp_js.html
- http://www.rcdevs.com/tokens/?type=software
- https://play.google.com/store/apps/details?id=uk.co.bitebullet.android.token&feature=search_result#?t=W251bGwsMSwxLDEsInVrLmNvLmJpdGV0aGVidWxsZXQuYW5kcm9pZC50b2tibiJd
- https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5nb29nbGUuYW5kcm9pZC5hcHBzLmF1dGhlbnRpY2F0b3IyIl0.