OpenVPN RADIUS MySQL/LDAP Howto

by croessner - 2010-11-26 11:40:28

http://www.roessner-network-solutions.com/beliebte-seiten-und-artikel/openvpn-radius-mysqlldap-howto/

Overview

This document describes how to setup a FreeRADIUS server. A MySQL server is used as backend and for the user accounting. OpenVPN and the radiusplugin from Ralf Lübben are used together as nas service.

I do not guarantee for anything in this howto. In my environment this setup is doing a great job here. So hopefully it will do the same for you.

[Update on 2008-10-03]:
This setup is also working with freeradius version 2.1.0, while this document originally was written for a 1.x version. Much of the structure has changed in this newer release, but you can apply this howto even for this version.

[Update on 2009-04-25]:
LDAP authentication coming soon. It is still on my personal wish list :-)

[Update on 2010-08-19]:
Added quick radiusplugin-build instructions at the bottom of this document

[Update on 2010-08-31]:
Finally got LDAP included :-)

[Update on 2010-10-15]:
Enabling tracebacks and ping backs for this side

Required software

The installation was done on Ubuntu Gutsy Gibbon and is still valid up to current Lucid Lynx (versions may differ at the moment):

- freeradius (1.1.6-2)
- freeradius-mysql (1.1.6-2)
- mysql-server-5.0 (5.0.45-1ubuntu2)
- openvpn (2.0.9-8)
- radiusplugin_v2.1_beta9.tar.gz (Please download separately)
- libgcrypt11-dev (1.2.4-2ubuntu2)

I act on the assumption that there is an already running MySQL server.

RADIUS-Server

After having isuccessfully nstalled freeradius and freeradius-mysql using aptitude (apt-get), you have to change the directory to /etc/freeradius.

radiusd.conf:

Please change the following variables under the section PROXY CONFIGURATION

proxy_requests = no

Please comment out any files-entry and as you can see, please enable the sql statements. The changes should look similar like this::

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

preacct {
preprocess
acct_unique
suffix
}

accounting {
detail
unix
radutmp
sql
}
```

For freeradisu 2.x in file /etc/freeradius/sites-enabled/default:

```
authorize {
sql
}
authenticate {
}
preacct {
acct_unique
}
accounting {
sql
}
session {
sql
}
post-auth {
```

```
}
pre-proxy {
}
post-proxy {
}
```

As you can see, you only require the sql statements and no others. Please give a feedback, if you require more information on freeradius 2.x configuration.

You do not need to change anything else in this configuration files. It keeps as it is.

clients.conf:

```
client 127.0.0.1 {
secret          = EinsupertollesSecret
shortname       = localhost
}
```

The secret should be a secret as far as possible. It will be required in a later configuration file below.

sql.conf:

```
sql {
driver = "rlm_sql_mysql"
server = "127.0.0.1"
login = "radius"
password = "MySQL-passowrd-see-next-paragraph"
radius_db = "radius"
...
}
```

MySQL

mysql -u root -h 127.0.0.1 -p

Please insert the following schema into MySQL:

```
zcat /usr/share/doc/freeradius/examples/mysql.sql.gz | \
mysql -u root -prootpass radius
```

```
mysql -u root -prootpass
mysql> GRANT ALL ON radius.* to radius@'127.0.0.1' IDENTIFIED BY 'Use the same password as in  sql.conf';
```

Next, some example entries:

```
mysql> select * from radcheck;
+----+------------+---------------+----+---------------+
| id | UserName   | Attribute     | op | Value         |
+----+------------+---------------+----+---------------+
|  1 | croessner  | Crypt-Password | := | XXXXXXXXXXXX |
+----+------------+---------------+----+---------------+
```

You can use the MySQL ENCRYPT() function to create the passwords.

```
mysql> select * from radgroupcheck;
+----+-----------+-----------+----+-------------+
| id | GroupName | Attribute | op | Value       |
+----+-----------+-----------+----+-------------+
|  1 | dynamic   | Auth-Type | := | Crypt-Local |
+----+-----------+-----------+----+-------------+

mysql> select * from radgroupreply;
+----+-----------+----------------------+----+-------------+
| id | GroupName | Attribute            | op | Value       |
+----+-----------+----------------------+----+-------------+
|  1 | dynamic   | Acct-Interim-Interval | =  | 60          |
+----+-----------+----------------------+----+-------------+

mysql> select * from radreply;
+----+------------+------------------+----+------------------------------+
| id | UserName   | Attribute        | op | Value                        |
+----+------------+------------------+----+------------------------------+
|  1 | croessner  | Framed-IP-Address | =  | 10.10.0.153                  |
|  2 | croessner  | Framed-Route     | =  | 192.168.3.0/24 10.10.0.2/32 1 |
+----+------------+------------------+----+------------------------------+
```

Short description:
After the user croessner as logged on, the IP 10.10.0.153 is assigned to his computer as a point-to-point connection with the endpoint IP 10.10.0.154. At the same time, the OpenVPN server manipulates its internal routing table and adds the network 192.168.3.0/24. If you wish to assign more than one route, you have to use the '+=' operator for any additional data set.

```
mysql> select * from usergroup;
+-----------+-----------+----------+
| UserName  | GroupName | priority |
+-----------+-----------+----------+
| croessner | dynamic   |        1 |
+-----------+-----------+----------+
```

I have to mention for the table shown here that the usage of the operators seems not to be really trivial. But you can find more information in /usr/share/doc/freeradius/rlm_sql.gz.

I explicitly use "Crypt-Password" entries in these examples. If this is not desired, you can use the attribute "Cleartext-Password". But doing so, you have to choose the value "Local" in the table "radgroupcheck".

You can find more information in the README under http://wiki.freeradius.org/SQL_HOWTO.

OpenVPN

RadiusPlugin

As of writing this howto, the freeradius plugin is not available as an Ubuntu package. Therefor you have to download and compile the source code. Please install the GNU compiler "g++" and "make". Simply a basic installation of tools, giving you the ability to compile C++ applications. Maybe the package "build-essential".

cd /usr/local/src/
wget http://www.nongnu.org/radiusplugin/radiusplugin_v2.0b_beta2.tar.gz
tar xvzf radiusplugin_v2.0b_beta2.tar.gz
cd /usr/local/src/radiusplugin_v2.0b_beta2

After that run "make".
The result is called radiusplugin.so.

cp /usr/local/src/radiusplugin_v2.0b_beta2/radiusplugin.so /etc/openvpn/

Please also copy the file radiusplugin.cnf from the directory /usr/local/src/radiusplugin_v2.0b_beta2 to /etc/openvpn.

The configuration should look something like this:

```
# The NAS identifier which is sent to the RADIUS server
NAS-Identifier=OpenVpn # The service type which is sent to the RADIUS server
Service-Type=5
# The framed protocol which is sent to the RADIUS server
Framed-Protocol=1
# The NAS port type which is sent to the RADIUS server
NAS-Port-Type=5
# The NAS IP address which is sent to the RADIUS server
NAS-IP-Address=127.0.0.1
# Path to the OpenVPN configfile. The plugin searches there for
# client-config-dir PATH   (searches for the path)
# status FILE                    (searches for the file, version must be 1)
# client-cert-not-required (if the option is used or not)
# username-as-common-name  (if the option is used or not)
OpenVPNConfig=/etc/openvpn/radiusvpn.conf
# Support for topology option in OpenVPN 2.1
# If you don't specify anything, option "net30" (default in OpenVPN) is used.
# You can only use one of the options at the same time.
# If you use topology option "subnet", fill in the right netmask, e.g. from
# OpenVPN option "--server NETWORK NETMASK"
#subnet=255.255.255.0
# If you use topology option "p2p", fill in the right network, e.g. from OpenVPN
# option "--server NETWORK NETMASK"
#p2p=10.10.0.1
######################### Ich benutze die Default Option
# Allows the plugin to overwrite the client config in client config file directory,
# default is true
overwriteccfiles=true
# Path to a script for vendor specific attributes.
# Leave it out if you don't use an own script.
# vsascript=/root/workspace/radiusplugin_v2.0.5_beta/vsascript.pl
# Path to the pipe for communication with the vsascript.
# Leave it out if you don't use an own script.
# vsanamedpipe=/tmp/vsapipe
# A radius server definition, there could be more than one.
# The priority of the server depends on the order in this file. The first one
# has the highest priority.
server
{
# The UDP port for radius accounting.
```

```
acctport=1813
# The UDP port for radius authentication.
authport=1812
# The name or ip address of the radius server.
name=127.0.0.1
# How many times should the plugin send the if there is no response?
retry=1
# How long should the plugin wait for a response?
wait=1
# The shared secret.
sharedsecret=Hier das Secret aus der client.conf des Radius-Servers
}
```

Point-to-Multipoint Server

Please setup a point-to-multipoint configuration. Tip: Use the easy-rsa-package, which you can install seperatly with aptitude:

i.e.:

```
cp -a /usr/share/doc/openvpn/examples/easy-rsa /etc
cd /etc/easy-rsa/2.0/
```

Edit the file vars and change the lines below, like described in the README.

```
source vars
./clean-all
./build-ca
./build-key-server server
./build-dh
```

Now you can create one or more client certificates:

```
./build-key cl1
```

```
cd keys
openvpn --genkey --secret ta.key
```

Please change to the directory /etc/openvpn

```
cd /etc/openvpn
mkdir ssl
cp -a /etc/easy-rsa/keys/{ca.crt,dh1024.pem,ta.key,server.crt,server.key} ssl/
```

Use an editor and put in the following sample configuration:

radiusvpn.conf:

```
# Which device
dev tun
fast-io

user nobody
group nogroup
```

```
persist-tun
persist-key

server 10.10.0.0 255.255.255.0
management 127.0.0.1 7505
float

username-as-common-name
client-config-dir ccd
client-to-client

push "redirect-gateway def1"
push "dhcp-option NTP 10.10.0.1"
push "dhcp-option DOMAIN lan"
push "dhcp-option DNS 10.10.0.1"

ping-timer-rem
keepalive 10 60

# Use compression
comp-lzo

# Strong encryption
tls-server
tls-auth ssl/ta.key 0
dh ssl/dh1024.pem
cert ssl/server.crt
key ssl/server.key
ca ssl/ca.crt

plugin /etc/openvpn/radiusplugin.so /etc/openvpn/radiusplugin.cnf

verb 3
mute 10

status /var/log/openvpn/status.log 1
log /var/log/openvpn/radiusvpn.log

mkdir /etc/openvpn/ccd
mkdir /var/log/openvpn
```

That´s it ;-) The server is ready to go. Now you can start the services freeradius, mysql and openvpn.

Afterwards you can configure the client(s). The following output is just an idea of how it could look like. Any further documentation can be found on the project website.

Client example

```
# Which device
dev tun
fast-io

persist-key
persist-tun
```

```
replay-persist radiusvpn.d/cur-replay-protection.cache

# Our remote peer
nobind
remote <HIER_REMOTE_ADRESSE_DES_OPENVPN_SERVERS> 1194

pull

# Use compression
comp-lzo

# Strong encryption
tls-client
tls-remote server
ns-cert-type server
tls-auth ssl/ta.key 1
cert ssl/common.crt
key ssl/common.key
ca ssl/ca.crt

verb 3
mute 10

auth-user-pass radiusvpn.d/auth-user-pass.conf

up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf

# log /var/log/openvpn.log
```

```
mkdir /etc/openvpn/radiusvpn.d
```

Change to the given directory and create the file auth-user-pass.conf. Please also refer to the openvpn manpage for the parameter --auth-user-pass.

Test it ... Be happy

Have fun - You can send me bug reports concerning the hoto to Christian Roessner <info@roessner-net.com> or if you like to, just give a comment.

Update 2010-08-19:

Extract the plugin archive:

```
>#  cd /usr/local/src
>#  wget "http://www.nongnu.org/radiusplugin/radiusplugin_v2.1_beta9.tar.gz"
>#  tar xvzf radiusplugin_v2.1_beta9.tar.gz
```

Building the radius plugin:

```
>#  cd radiusplugin
>#  make
>#  cp radiusplugin.so /etc/openvpn/plugins/
```

Update 2010-08-31:

LDAP for authorization and authentication

Instead of using MySQL for authorization and authentication, you can bind FreeRADIUS at an LDAP server. I have not done this with OpenVPN as a NAS yet, but with pppoe-server (rp-pppoe) and the steps should be nearly the same. Here is what I have done.

To use LDAP with freeradius, you need to install freeradius-ldap and slapd.

```
authorize {
preprocess
files
sql
ldap
expiration
logintime
}
authenticate {
Auth-Type LDAP {
ldap
}
}
preacct {
preprocess
acct_unique
suffix
files
}
accounting {
sql
}
session {
sql
}
post-auth {
ldap
exec
}
pre-proxy {
}
post-proxy {
}
```

Notice: You also need the **files** module, else you can not have LDAP looking up profiles for reply-items. At the moment I do not know, if there is another way for looking up GroupName stuff. Maybe someone else might give a hint here ;-)

Modify the users file like this (example):

```
DEFAULT Ldap-Group == disabled, Auth-Type := Reject
Reply-Message = "Account disabled.  Please call the helpdesk.",
```

```
Fall-Through = no

DEFAULT Ldap-Group == flat10000, User-Profile :=
"uid=flat10000,ou=profiles,ou=radius,ou=wl,dc=example,dc=org"
Fall-Through = no

DEFAULT Auth-Type := Reject
Reply-Message = "Please call the helpdesk."
```

The ldap module configuration for freeradius might look like this:

```
ldap {
server = "wl00.wl.example.org" # Insert your exact FQDN here, if using TLS
identity = "cn=proxyuser,dc=example,dc=org"
password = YOUR-LDAPY-PROXYUSER-PW-HERE
basedn = "ou=wl,dc=example,dc=org"
filter = "(uid=%{%{Stripped-User-Name}:-%{User-Name}})"
base_filter = "(objectclass=radiusprofile)"
ldap_connections_number = 5
timeout = 4
timelimit = 3
net_timeout = 1
tls {
start_tls = yes
cacertfile  = /ca/cacert_org.crt # I use certificates signed by http://www.cacert.org
require_cert = "demand"
}
dictionary_mapping = ${confdir}/ldap.attrmap
password_attribute = userPassword
edir_account_policy_check = no
groupname_attribute = radiusGroupName
groupmembership_filter = "(&(uid=%{%{Stripped-User-Name}:-%{User-Name}})(objectclass=radiusprofile))"
compare_check_items = no
}
```

Add the freeradius-schema for LDAP to the slapd.conf (or include it in slapd.d).

A sample init.ldif is shown here:

```
dn: dc=example,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
dc: example
o: MyCompany

dn: ou=wl,dc=example,dc=org
objectClass: organizationalUnit
objectClass: top
ou: wl

dn: ou=users,ou=wl,dc=example,dc=org
objectClass: organizationalUnit
```

```
objectClass: top
ou: users

dn: ou=radius,ou=wl,dc=example,dc=org
objectClass: organizationalUnit
objectClass: top
ou: radius

dn: ou=profiles,ou=radius,ou=wl,dc=example,dc=org
objectClass: organizationalUnit
objectClass: top
ou: profiles

# This sample is from PPPoE and shows some vendor specific attributes
dn: uid=flat10000,ou=profiles,ou=radius,ou=wl,dc=example,dc=org
objectClass: radiusObjectProfile
objectClass: top
objectClass: radiusprofile
uid: flat10000
cn: flat10000
radiusReplyItem: Acct-Interim-Interval := 360
radiusReplyItem: RP-Downstream-Speed-Limit := 10240
radiusReplyItem: RP-Upstream-Speed-Limit := 10240
radiusIdleTimeout: 3600
radiusSessionTimeout: 86400
radiusSimultaneousUse: 1

dn: cn=proxyuser,dc=example,dc=example
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: proxyuser
userPassword: {SSHA}***************
description: LDAP administrator (read-only)

dn: uid=wl00000000,ou=users,ou=wl,dc=example,dc=org
objectClass: inetOrgPerson
objectClass: radiusprofile
uid: wl00000000
cn: Christian Roessner
sn: Roessner
givenName: Christian
l: Cityname_here
postalCode: Zip_code_here
postalAddress: Foobar street 4711
homePhone: +49 000 00000000
mail: sample@example.org
userPassword: Test123West
description: Testuser
radiusGroupName: flat10000
```

Notice: Maybe you see that I am using cleartext passwords. This differs from using MySQL as source for storing users/pws. I do not see this as a security provlem.

I have configured LDAP to have a proxyuser that has access rights to all data with read-only support.

Here is my sample slapd.conf:

```
# /etc/ldap/slapd.conf

include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema
include         /etc/ldap/schema/freeradius.schema # You can find it in the doc folder somewhere in freeradius

argsfile        /var/run/slapd/slapd.args
pidfile         /var/run/slapd/slapd.pid

modulepath      /usr/lib/ldap
moduleload      back_hdb.la

loglevel        256

# Sample security restrictions
#       Require integrity protection (prevent hijacking)
#       Require 112-bit (3DES or better) encryption for updates
#       Require 63-bit encryption for simple bind
security        ssf=1 update_ssf=112 simple_bind=64

TLSCACertificateFile    /ca/cacert_org.crt
TLSCertificateFile      /ca/newcert.pem
TLSCertificateKeyFile   /ca/newkey.pem

database        frontend

# Sample access control policy:
#       Root DSE: allow anyone to read it
#       Subschema (sub)entry DSE: allow anyone to read it
#       Other DSEs:
#               Allow self write access
#               Allow authenticated users read access
#               Allow anonymous users to authenticate
#       Directives needed to implement policy:
access to dn.base=""
by * read
access to dn.base="cn=Subschema"
by * read
access to *
by self write
by users read
by anonymous auth

database        config
rootdn          cn=config
rootpw          {SSHA}*****************

database        hdb
suffix          dc=example,dc=org
rootdn          cn=admin,dc=example,dc=org
rootpw          {SSHA}*****************
```

```
directory       /var/lib/ldap
index           objectClass eq
# ... More indexes where added with Apache-Directory-Studio and not listed here

access to attrs=userPassword,shadowLastChange
by self write
by dn.exact="cn=proxyuser,dc=example,dc=org" read
by anonymous auth
by * none

access to *
by dn.exact="cn=proxyuser,dc=example,dc=org" read
by users read
by * none
```

After finishing, you can delete everything from the MySQL server concerning users. The only table that will still be used is the radacct table. All the other tables are empty. But you also can store users in both servers. Storing one user in both is a bad idea ;-)

See a final radtest here:

```
radtest wl00000000 PW_for_wl00000000 127.0.0.1 0 The_Client_PW_for_radius
Sending Access-Request of id 215 to 127.0.0.1 port 1812
User-Name = "wl00000000"
User-Password = "PW_for_wl00000000"
NAS-IP-Address = 127.0.1.1
NAS-Port = 0
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=215, length=62
Idle-Timeout = 3600
Session-Timeout = 86400
Acct-Interim-Interval = 360
RP-Downstream-Speed-Limit = 10240
RP-Upstream-Speed-Limit = 10240
```

And LDAP sample output:

```
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 fd=15 ACCEPT from IP=127.0.1.1:54769 (IP=0.0.0.0:389)
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 op=0 EXT oid=1.3.6.1.4.1.1466.20037
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 op=0 STARTTLS
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 op=0 RESULT oid= err=0 text=
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 fd=15 TLS established tls_ssf=128 ssf=128
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 op=1 BIND dn="cn=proxyuser,dc=example,dc=org" method=128
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 op=1 BIND dn="cn=proxyuser,dc=example,dc=org" mech=SIMPLE
ssf=0
Aug 30 17:01:21 wl00 slapd[5100]: conn=2126 op=1 RESULT tag=97 err=0 text=
...
...
Aug 30 19:08:42 wl00 slapd[5100]: conn=2126 op=15 SRCH base="ou=wl,dc=example,dc=org" scope=2 deref=0
filter="(uid=wl100001)"

Aug 30 19:08:42 wl00 slapd[5100]: conn=2126 op=15 SRCH attr=radiusNASIpAddress radiusExpiration acctFlags
dBCSPwd sambaNtPassword sambaLmPassword ntPassword lmPassword radiusCallingStationId radiusCal
ledStationId radiusSimultaneousUse radiusAuthType radiusCheckItem radiusTunnelPrivateGroupId
```

radiusTunnelMediumType radiusTunnelType radiusReplyMessage radiusLoginLATPort radiusPortLimit radiusFramedA
ppleTalkZone radiusFramedAppleTalkNetwork radiusFramedAppleTalkLink radiusLoginLATGroup radiusLoginLATNode radiusLoginLATService radiusTerminationAction radiusIdleTimeout radiusSessionTimeout radiusCl
ass radiusFramedIPXNetwork radiusCallbackId radiusCallbackNumber radiusLoginTCPPort radiusLoginService radiusLoginIPHost radiusFramedCompression radiusFramedMTU radiusFilterId radiusFramedRouting radi
usFramedRoute radiusFramedIPNetmask radiusFramedIPAddress radiusFramedProtocol radiusServiceType radiusReplyItem userPassword sasdefaultloginsequence
Aug 30 19:08:42 wl00 slapd[5100]: conn=2126 op=15 SEARCH RESULT tag=101 err=0 nentries=1 text=

Aug 30 19:08:42 wl00 slapd[5100]: conn=2126 op=16 SRCH base="uid=flat10000,ou=profiles,ou=radius,ou=wl,dc=example,dc=org" scope=0 deref=0 filter="(objectClass=radiusprofile)"
Aug 30 19:08:42 wl00 slapd[5100]: conn=2126 op=16 SRCH attr=radiusNASIpAddress radiusExpiration acctFlags dBCSPwd sambaNtPassword sambaLmPassword ntPassword lmPassword radiusCallingStationId radiusCal
ledStationId radiusSimultaneousUse radiusAuthType radiusCheckItem radiusTunnelPrivateGroupId radiusTunnelMediumType radiusTunnelType radiusReplyMessage radiusLoginLATPort radiusPortLimit radiusFramedA
ppleTalkZone radiusFramedAppleTalkNetwork radiusFramedAppleTalkLink radiusLoginLATGroup radiusLoginLATNode radiusLoginLATService radiusTerminationAction radiusIdleTimeout radiusSessionTimeout radiusCl
ass radiusFramedIPXNetwork radiusCallbackId radiusCallbackNumber radiusLoginTCPPort radiusLoginService radiusLoginIPHost radiusFramedCompression radiusFramedMTU radiusFilterId radiusFramedRouting radi
usFramedRoute radiusFramedIPNetmask radiusFramedIPAddress radiusFramedProtocol radiusServiceType radiusReplyItem userPassword sasdefaultloginsequence
Aug 30 19:08:42 wl00 slapd[5100]: conn=2126 op=16 SEARCH RESULT tag=101 err=0 nentries=1 text=

If you run the ldap- and freeradius server on the same machine, you also could forget about using tls and use a unix socket instead (/etc/freeradius/module/ldap: server="ldapi://%2fvar%2frun%2fslapd%2fldapi"). This works with ssf from slapd.conf aswell. I use ldapi and tls, so I can manage LDAP from remote with Apache Directory Studio and have a working setup, even I forgot to renew the server certificate ;-)

I know the part binding freeradisu to an ldap might be not as good as the first part of this howto, but I am short in time ;-) Hope it works for you.