FreeRADIUS Database Connection

Best Practice Document

Produced by CSC/Funet led working group MobileFunet

Wenche Backman-Kamila (CSC), Tuukka Vainio (University of Turku), Miika Räisänen (University of Oulu) and Thomas Backa (Åbo Akademi University)

18.1.2013

Document No: GN3-NA3-T4-freeradius-database-connection
Version / date: 18.1.2013
Original language: Finnish
Original title: “FreeRADIUS tietokantaliitos”
Original version / date: 1.0 of 4.6.2012
Contact: Wenche.Backman-Kamila at csc.fi

CSC/Funet bears responsibility for the content of this document. The work has been carried out by a CSC/Funet led working group MobileFunet as part of a joint-venture project within the HE sector in Finland.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 238875, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3)'.

Table of Contents

Executive Summary

These instructions are a part of the MobileFunet eduroam guidelines, which contain best practices to assist providing wireless network services on campuses. MobileFunet is a working group consisting of experts from CSC, Finnish universities and polytechnics.

These instructions describe how to connect a FreeRADIUS server to an external user databases and directories, the use of which is practically mandatory for minimizing administrative work. The instructions continue the configuration of a FreeRADIUS server set up according to MobileFunet's FreeRADIUS Configuration BPD [1].

The main author of these instructions is Wenche Backman-Kamila, with the exception of sections ‘Taking Several Databases into Consideration’ and ‘EAP-TLS Authentication’, which were written by Tuukka Vainio.

1 Connecting to Active Directory

1.1 Samba and Kerberos

Samba [2] and Kerberos are required for connecting a FreeRADIUS server to an Active Directory (AD) user database. In the case of RHEL, you can either install the distribution's own version of Samba using the yum command, or install the latest version of Samba by downloading the necessary packages from the Internet and using the rpm command. As long as the distribution's own version is newer than 3.0.0, we recommend using yum, as it makes update management easier.

1.1.1 Option 1 – Installing Samba with yum

You can find which version of Samba and the Winbind programme installed together with Samba is included in the distribution with the yum info command (yum info *samba* and yum info *winbind*). Samba and Winbind are installed using the yum install command. In addition to the software itself, we recommend that you also install the included documentation files samba*-doc.

1.1.2 Option 2 – Downloading Samba packages and installing them with rpm

Begin by downloading Samba from the enterprisesamba.com link [3] on the downloads page [4], for example. Download the following packages as a minimum:

- *samba3*.rpm, which requires*
  - *libwbclient*.rpm*
  - *samba3-client*.rpm*
- *samba3-winbind*.rpm*

Next, install the packages by running the following commands: *rpm -ihv (or -Uhv) libwbclient*.rpm, rpm -ihv* (or *-Uhv*) *samba3-client*.rpm, rpm -ihv* (or *-Uhv*) *samba3*.rpm* and *rpm -ihv* (or *-Uhv*) *samba3-winbind*.rpm* . It may also be sensible to install the following packages: *samba3-utils*.rpm, samba3-debuginfo*.rpm* and *samba3-doc*.rpm* in the same way. *samba3-doc* may require additional packages, which you can find here [5].

1.1.3 Samba configuration

First, check that Samba and Winbind will start by running the following commands: */etc/init.d/smb* start and */etc/init.d/winbind* start.

Once Samba has been installed, configure the /etc/samba/smb.conf file. You can find a template, for example, in the Samba Wiki [6], but you can also find an example of a configuration file tailored to meet the needs of authentication only in file attachment [7] including explanations. You can find more information on the different parameters in the samba3-doc package's smb.conf.5.html file in the ..../samba3/htmldocs/manpages/-directory. You can test the settings by running *testparm smb.conf*.

1.1.4 Kerberos

Next, define the parameters of the Kerberos application. It is sufficient that the following packages have been installed on the server:

- *krb5-workstation*
- *krb5-libs*

Kerberos parameters are defined in the /etc/krb5.conf file. You can find an example of the Kerberos parameters in the Samba Wiki [6]. See the file attachment [8] for sample parameters for connecting FreeRADIUS and AD. Note that the configuration file is case-sensitive.

As the operation of Kerberos is dependent on the precise time, you should check at this point that the server uses the same NTP servers as AD. You can define the NTP servers in the etc/ntp.conf file. Define Funet's NTP servers ntp1.funet.fi and ntp2.funet.fi as follows:

```
restrict 193.166.5.177 mask 255.255.255.255 nomodify notrap noquery
server 193.166.5.177
restrict 193.166.5.197 mask 255.255.255.255 nomodify notrap noquery
server 193.166.5.197
```

1.1.5 Connecting to Active Directory

Next, launch Samba and Winbind using the commands */etc/init.d/smb start* and */etc/init.d/winbind start*. Connect the server to AD by running the command *net ads join –S <domain_controller> –U <administrator account> –W windows*. You should then restart Winbind: */etc/init.d/winbind restart*. You can test whether the connection was successful with all user accounts defined in AD, for example as follows: *wbinfo -a username*. Enter your password and ensure that the test ends in the notification authentication succeeded. Samba's ntlm_auth application will be used with FreeRADIUS, so it should be tested as well:

```
ntlm_auth --username username --password
password:
NT_STATUS_OK: Success (0x0)
```

Before beginning the configuration of FreeRADIUS, you must grant the radiusd user rights to use Winbind. You can do this by modifying the rights to the winbindd_privileged directory. If wbpriv has been defined as the group user of the winbindd_privileged directory, we recommend adding the radius user to the wbpriv group, for example with the useradd command (*useradd -G wbpriv radiusd*). If the wbpriv group is missing, you can directly add radiusd as a group user of the winbindd_privileged directory with the command *chown root:radiusd /var/lib/samba/winbindd_privileged*.

Finally, you should check that the group has read and execution rights to the directory. If not, add them with the command *chmod 750 /var/lib/samba/winbindd_privileged*.

1.2 Configuration

In AD, user passwords are stored in the NTHASH format, and any EAP and EAP-type methods are determined according to this table [9]. Usually, when FreeRADIUS is connected to AD, the PEAP-MSCHAPv2, TTLS-MSCHAPv2 and TTLS-PAP methods are supported. Before starting the actual configuration, ensure that your realm has been defined in the **proxy.conf** file as an exception (using csc.fi as an example).

```
realm csc.fi {
}
```

Authentication is handled through the ntlm_auth application for all methods, but the function call is slightly different, depending on the method.

1.2.1 Configuration to support the TTLS-PAP method

You can support the TTLS-PAP method by first creating a file called *ntlm_auth* in the /etc_raddb/modules directory and adding the following lines into it:

```
exec ntlm_auth{
wait = yes
program = "/usr/bin/ntlm_auth --request-nt-key --domain=windows --
username=%{Stripped-User-Name} --password=%{User-Password}"
}
```

Change the value of the --domain parameter if you wish to use your own specific domain here (--domain=MYDOMAIN).

Next, add the lines marked with #add into the eduroam-inner-tunnel file in the /etc/raddb/sites-available directory:

```
authorize {
 auth_log
 files
 suffix
 mschap
 #If mschap returns noop, set Auth-Type value to ntlm_auth so that TTLS-PAP
 #will work with AD.
```

```
 #mschap returns ok to TTLS-MSCHAPv2 and noop to TTLS-PAP. mschap also returns
 #noop, if the inner method is of the EAP type, i.e. PEAP-MSCHAPv2 or
 #TTLS-(EAP-MSCHAPv2).
 #The inner method of PEAP must always be EAP-type.
 if (noop) { #add
   update control{ #add
         Auth-Type := ntlm_auth #add
         } #add
 } #add
 pap
 eap {
   ok = return
 }
 #If eap returns updated, the authentication method (Auth-Type) is EAP and the
 #earlier defined ntlm_auth can be removed. eap returns updated if the inner
 #method is EAP-type, i.e.PEAP-MSCHAPv2 or TTLS-(EAP-MSCHAPv2).
 #eap returns noop to TTLS-MSCHAPv2 and TTLS-PAP.
 #If this if block is deleted, authentication will still work, but two
 #authentication methods will then be defined in cases where the inner
 #method is EAP-type: EAP and ntlm_auth.
 if (updated) { #add
   update control{ #add
         Auth-Type -= ntlm_auth #add
    } #add
 } #add
}

authenticate {
 Auth-Type PAP{
   pap
 }
 Auth-Type MS-CHAP{
   mschap
 }
 Auth-Type ntlm_auth{ #add
   ntlm_auth #add
 } #add
 # Allow EAP authentication.
 eap
}
...
```

When the eduroam inner-tunnel file is modified in this way, PEAP-MSCHAPv2, TTLS-MSCHAPv2 and TTLS-EAP-MSCHAPv2 can be supported as well.

When configuring FreeRADIUS, we recommend that you also familiarise yourself with its own processing language (FreeRADIUS Processing un-language). You can study the processing language by reading the manual included with the server. Open it by entering *man unlang* on the command line.

1.2.2 Configuration for supporting the PEAP-MSCHAPv2, TTLS-MSCAPv2 and TTLS-(EAP-MSCHAPv2) methods

In the case of the PEAP-MSCHAPv2, TTLS-MSCHAPv2 and TTLS-(EAP-MSCHAPv2) methods, the ntlm_auth application is called from the mschap module when you wish to use AD for authentication. Edit the last line of the /etc/raddb/modules/mschap file as follows:

```
ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --username=%{%{Stripped-User-Name}:-%{User-Name:-None}} --domain=%{%{mschap:NT-Domain}:-windows} --challenge=%mschap:Challenge:-00} --nt-response=%{mschap:NT-Response:-00}"
```

Delete the # at the beginning of the line. This definition sets the domain value to 'windows', if the NT-Domain attribute is not set elsewhere. If you wish to use your own domain here, you should add the definition --domain = MYDOMAIN.

You should also set the other parameters of the mschap module at the same time:

```
use_mppe = yes
require_encryption = yes
require_strong = yes
with_ntdomain_hack = no
```

1.2.3 Other

If you do not wish to hardcode the AD domain, it can also be defined, for example, based on the realm. The definition can be made, for example, in the authorize block of the /etc/raddb/sites-enabled/eduroam-inner-tunnel file:

```
authorize {
...
 # Strip domain to ntlm_auth attribute
 if ("%{User-Name}" =~ /@myorganisation.fi$/i){
 update request {
   Stripped-User-Domain = 'my-windows-domain'
   }
 }
}
```

In this case, Stripped-User-Domain is set as the value of the domain parameter of the ntlm_auth call:

```
--domain=%{Stripped-User-Domain}
```

Stripped-User-Domain must be defined before it can be used. Define it by adding the following line in the /etc/raddb/dictionary file:

```
ATTRIBUTE Stripped-User-Domain 3009 string
```

1.3 Experiences

- Samba version 3.5.10-114(.el6.i686) works well with Windows 2008 R2.
- If you are using SELinux, you might need additional configuration.
- We have noticed that the Winbind demon may crash occasionally, apparently due to a memory leak. You can circumvent this problem by restarting Winbind once a week if updating Winbind does not help.

2 Connecting to an LDAP database (OpenLDAP)

You can connect a FreeRADIUS server to an LDAP database through the rlm_ldap module. If the module was not installed during the FreeRADIUS installation, do it now by running the command *rpm -Uhv freeradius-ldap-2.1.3-1.x86_64.rpm* in the /usr/src/redhat/RPMS/x86_64 directory.

The FreeRADIUS server is connected to the LDAP database by editing the configuration files as follows:

In *proxy.conf* define the realm authenticating against LDAP as an exception:

```
realm ldapdomain.fi {
}
```

In *eduroam-inner-tunnel* add a suffix definition in the authorize block to remove the realm section, and an ldap definition. Without the suffix definition, the database search would be performed for the user *username@ldapdomain.fi*, but with the suffix definition, the search will only be performed for the user *username*.

```
authorize {
 auth_log
 files
 suffix #add
 ldap #add
 mschap
 pap
 eap {
   ok = return
 }
}
```

In *etc/raddb/modules/ldap* define the LDAP server and the properties of its database in this file. If the LDAP server is located on the same physical server as the FreeRADIUS server, *localhost* is a sufficient definition. If not, define the server name here: *ldapserver.realm.fi*. You can also define the LDAP database user; that username and password will then be used to connect to the database. If the LDAP database user is left undefined, the database is connected to using the authenticating user's username and password, which is the better alternative considering information security. For this reason, the information for the LDAP database user is below in the comments. The Organisation Unit (ou) of the users is also defined.

```
ldap {
 server = "localhost" #or "ldapserver.realm.fi"
 #identity = "cn=Manager,dc=ldaprealm,dc=fi"
 #password = ldaptestsecret
 basedn = "ou=Funetpeople,dc=ldaprealm,dc=fi"
 filter = "(uid=%{%{Stripped-User-Name}:-%{User-Name}})
 ...
}
```

In the *etc/raddb/ldap.attrmap* file, define which RADIUS attribute matches which LDAP attribute. If the user passwords in the database are in clear text, define *Cleartext-Password*, and if they are in NTHASH format, define *NT-Password*.

```
checkItem User-Name uid
checkItem Cleartext-Password userPassword
#OR
   checkItem NT-Password userPassword
```

3 Taking more than one database into consideration

If FreeRADIUS is connected to more than one external user database, you should note that the failure of even one authentication module will, by default, cause the entire authentication to fail. By grouping the authentication modules, authentication can be performed in a failure-tolerant fashion:

```
server eduroam {
 authorize {
   auth_log
   suffix
   group {
         eap {
               fail = 1
               ok = return
         }
         ldap {
               fail = 1
               ok = return
         }
         sql {
               fail = 1
               ok = return
         }
   }
 }
 ...
```

The first module returning OK will end the authentication process. If none of the modules returns OK, an Access-Reject response is returned to the user.

4 EAP-TLS authentication

In addition to FreeRADIUS configuration, EAP-TLS authentication using certificates requires a public key management system; in a Windows environment, an easy alternative is Windows Server Active Directory Certificate Services [10] (Active Directory Certificate Services Step-by-Step Guide [11]). These instructions do not help in deploying PKI; they describe the changes required in order to connect FreeRADIUS to a PKI environment. The assumption is that the FreeRADIUS server already has a working eduroam configuration.

The PKI environment's root certificate (UniversityOfCityCA.crt below) is required, and possibly the server certificates granted to the FreeRADIUS servers, if you do not wish to use the certificates used in eduroam for the identification of the RADIUS server.

In principle, FreeRADIUS should identify which EAP method is used, but the easiest way is to copy a new eap instance (eap-tls) to the end of **eap.conf**:

```
eap {
    # EAP configuration that works with eduroam.
    ...
}
eap eap-tls {
    default_eap_type = tls
    timer_expire = 60
    ignore_unknown_eap_types = no
    cisco_accounting_username_bug = no
    max_sessions = 4096
    tls {
        certdir = ${confdir}/certs
        cadir = ${confdir}/certs
        private_key_file = ${certdir}/radius.univ.fi.key
        certificate_file = ${certdir}/radius.univ.fi-bundle.crt
        CA_file = ${cadir}/UniversityOfCityCA.crt
        dh_file = ${certdir}/dh
        random_file = ${certdir}/random
        cipher_list = "HIGH:!ADH:!MD5"
        fragment_size = 1024
        include_length = yes
        check_crl = no
        check_cert_issuer = "/DC=fi/DC=yo/CN=University of City"
    }
}
```

The5 essential changes to the variables are enlisted in the Table 1.

Table 1: EAP settings for the certificate alternatives.

Variable	With eduroam certificates	With certificates of own CA
default_eap_type	tls	
private_key_file	The key used in eduroam	The key used with own CA
certificate_file	the eduroam server certificate and certificate chain (e.g. TERENA-bundle)	server certificate of own CA
CA_file	the root certificate of own CA, as the client certificates are checked against this	
check_cert_issuer	Openssl x509 –noout –in UniversityOfCityCA.crt –issuer	

If you wish to check the client certificates against revocation, we recommend using OCSP (support was included in FreeRADIUS 2.1.11, sample configuration in the FreeRADIUS Wiki [12]). CRL checks are not handy, as a check made by FreeRADIUS itself requires a local CRL file and the restart of FreeRADIUS should the CRL file change, and OpenSSL called as an external command also requires a local CRL file, as it is unable to check the file over HTTP (see FreeRADIUS and CRLs [13]).

If you cannot use a dedicated virtual server with EAP-TLS (for example, because the same WLAN controller is in use), you can call the eap-tls instance, for example, in connection with the authentication of the other SSID. In such a case, make the following changes to the **sites-available/eduroam** file:

```
server eduroam {
    ...
    authenticate {
      Auth-Type eap {
       if (Aruba-Essid-Name == "YO Staff") { # or another way of getting the  SSID
           eap-tls
       } else {
           eap
       }
     }
     ...
```

5.1 The EAP-TLS settings of a Windows 7 supplicant

On the Security tab:

- Security type: ***WPA2-Enterprise***
- Encryption type: ***AES***
- Choose a network authentication method: ***Microsoft: Smart Card or other certificate***
- Remember my credentials for this connection each time I'm logged on
- After clicking the Settings button:
    - When connecting: ***Use a certificate on this computer***
        - Use simple certificate selection
    - Validate server certificate
        - Servers: ***radius.univ.fi;radius2.univ.fi***
        - Trusted Root Certification Authorities: ***University of City CA*** (or the trust relationship required by TERENA certificates)
        - Do not prompt the user to authorise new servers or trusted certification
- After clicking the Advanced settings button:
    - On the 802.1X settings tab:
        - Specify authentication mode: ***Computer authentication***

6 Comments

You can also connect Linux to Active Directory by following the instructions [14]. In RHEL6, you need to install the samba-common and samba-winbind packages, add the radiusd user into the wbpriv group and launch the Winbind demon. --Tuukka Vainio/TY

References

[1]	FreeRADIUS configuration BPD https://info.funet.fi/wiki/display/avoin/FreeRADIUSen+konfigurointi
[2]	Samba project home page http://www.samba.org/
[3]	Samba packages for enterprise ready Linux distributions http://enterprisesamba.com/
[4]	Samba project download page http://www.samba.org/samba/download/
[5]	RPM PBone Search tool for users of Linux RPM based distributions http://rpm.pbone.net/
[6]	Samba project, configuring Linux Login's and Samba Shares to authenticate against AD http://wiki.samba.org/index.php/Samba_&_Active_Directory
[7]	Example configuration file for Samba https://info.funet.fi/wiki/download/attachments/5702130/smb.conf
[8]	Kerberos configuration file https://info.funet.fi/wiki/download/attachments/5702130/krb5.conf
[9]	Deploying RADIUS:The book, Protocol and Password Compatibility http://deployingradius.com/documents/protocols/compatibility.html
[10]	Microsoft Windows Server, Active Directory Certificate Services http://technet.microsoft.com/en-us/windowsserver/dd448615
[11]	Microsoft Windows Server, Active Directory Certificate Services Step-by-Step Guide http://technet.microsoft.com/en-us/library/cc772393%28WS.10%29.aspx
[12]	FreeRADIUS documentation wiki http://wiki.freeradius.org/Eap.conf/eaedb27b545ae8177cac6744ab09437c72594c1c
[13]	Techno Bobbins, FreeRADIUS and CRLs http://sites.google.com/site/techbobbins/home/articles/freeradius-and-crls
[14]	Deploying RADIUS:The book, Configuring Authentication with Active Directory http://deployingradius.com/documents/configuration/active_directory.html

Glossary

AD	Active Directory
AES	Advanced Encryption Standard
CRL	Certificate Revocation List
EAP-TLS	Extensible Authentication Protocol-Transport Layer Security
LDAP	Lightweight Directory Access Protocol
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OU	Organisation Unit
PEAP-MSCHAP	Protected Extensible Authentication Protocol - Microsoft Challenge Handshake Authentication Protocol
SELinux	Security-Enhanced Linux
SSID	Service Set Identification
TTLS-MSCHAP	Tunneled Transport Layer Security – Microsoft Challenge Handshake Authentication Protocol
TTLS-PAP	Tunneled Transport Layer Security - Password Authentication Protocol