FreeRADIUS
Tutorial for AD integration

Version 1.2

Charles Schwartz
Network Security Engineer

Index

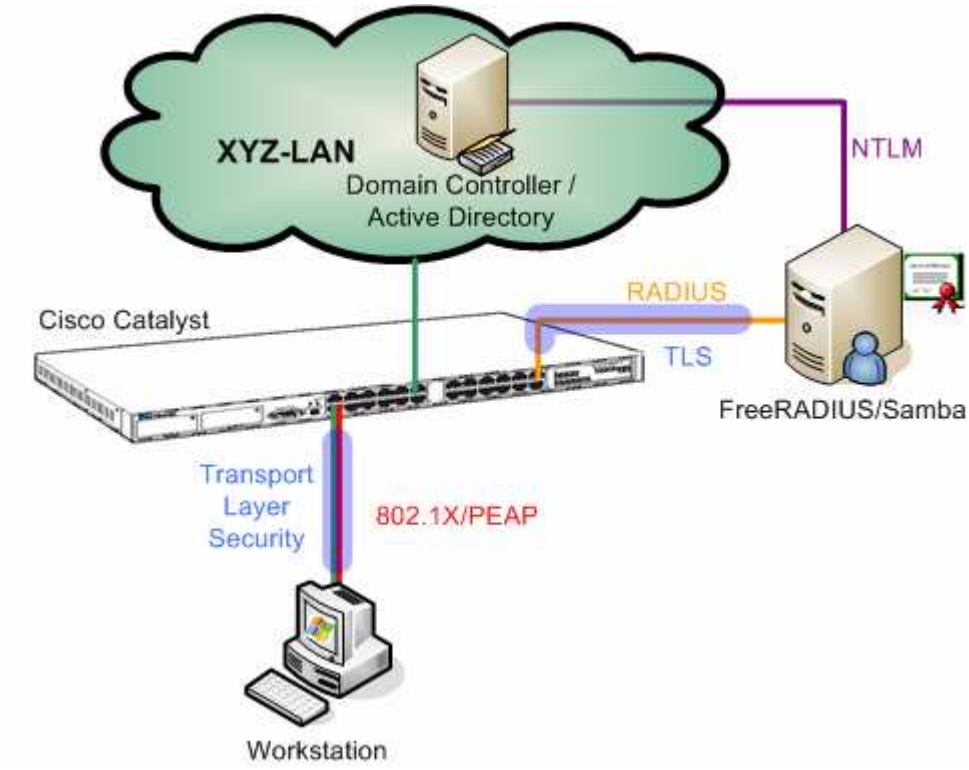
Introduction

This document describes how to set up FREERADIUS server in order to authenticate Windows XP network users transparently against Active Directory.
It is a step by step 'quick&dirty' guide to configure FREERADIUS server, network access points and WindowsXP supplicants.

I am not going to explain into detail each command or configuration line provided in this tutorial.
If you are not familiar with some expressions, please use Google to find out some more about them or just red the manuals that come with the software/hardware.

I Principles

FREERADIUS offers authentication via port based access control. A user can connect to the network only if its credentials have been validated by the authentication server. User credentials are verified by using special authentication protocols which belong to the 802.1X standard.



Refer to the graphic. Network access is only granted to the workstation if the user credentials have been authenticated by the FREERADIUS server. Otherwise the switch port will be down for any network traffic. The RADIUS server is allowed to contact the domain controller for user authentication.
Although the switch port is down, the workstation can communicate with the RADIUS server via an authentication protocol.
The RADIUS server is able to check on the domain controller if the user exists and if its password is correct. If this is the case, the RADIUS server tells the switch to open the port and the user will get access to the network.

II Prerequisites

The following components are required to install the access control solution:

- A Linux server
- FREERADIUS 1.0.x
- Samba 3.0.x
- Openssl

- Cisco Catalyst Switch

- Windows XP clients (Win2k is not supported!)

The Linux distribution used is this context was Fedora Core 3.

III Set up the Linux server

Linux must be configured in order to belong to a Windows domain. This is done by using the Samba file server which offers several interesting tools.
The goal is not to create a Samba file server but only to use some tools which come with this server.

Samba server contains among others the following components:

- ***Winbind***, a daemon which permits connectivity to Windows –NT environment.
- ***Ntlm_auth***, a tool which uses winbind for evaluating NTLM (NT Lan Manager) requests. This tool allows verifying user credentials on the domain controller and returns either a success or an error message.

Please have a look at your Linux box and check if Samba is already installed.

```
[root@radiussrv1]# rpm –qa | grep samba
```

Find the file `smb.conf` and open it with your preferred editor.

The file must contain the following lines:

In the `[global]` section

```
   # workgroup = NT-Domain-Name or Workgroup-Name
     workgroup = XYZDOM  //the name of your domain

   # Security mode. Most people will want user level
   # security. See security_level.txt for details.
     security = ads


#==================== Share Definitions ====================
   idmap uid = 16777216-33554431
   idmap gid = 16777216-33554431
   template shell = /bin/bash
   winbind use default domain = no
   password server = XYZSRV.XYZ-COMPANY.COM //your AD-server
   realm = XYZ-COMPANY.COM  //your realm
```

Verify the following lines in the [homes] section

```
comment = Home Directories
browseable = no
writable = yes
```

Next, find the file krb5.conf.
Normally it should be found in /etc/krb5.conf.

Edit this file with the following information: (Watch out for case sensitivity)

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = EXAMPLE.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false

[realms]
 EXAMPLE.COM = {
  kdc = kerberos.example.com:88
  admin_server = kerberos.example.com:749
  default_domain = example.com
 }

 XYZ-COMPANY.COM = {
  kdc = XYZSRV.XYZ-COMPANY.COM
 }

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM

[kdc]
 profile = /var/kerberos/krb5kdc/kdc.conf

[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
   krb4_convert = false
 }
```

Edit the file /etc/nsswitch.conf and add « winbind » at the end of each line shown below:

```
passwd:     files winbind
shadow:     files winbind
group:      files winbind

protocols:  files winbind

services:   files winbind

netgroup:   files winbind

automount:  files winbind
```

Restart the machine.

Verify if the Samba service is running by typing:
```
ps –ef | grep nmbd
ps –ef | grep smbd
```

Execute the following command line (you must be connected as root)

```
net join –U Administrator
```

« Administrator » is the name of the domain controller admin.
Enter your password when prompted.
If everything works fine, the Linux server has been registered to the Windows domain.

Verify now if the winbindd daemon is running:
```
~#ps –ef | grep winbindd
```

Try next if you can authenticate a user from the domain:

```
~#wbinfo –a user%password
```

The output should be something like the following:

```
[root@radiussrv1]# wbinfo –a CHSchwartz%mypassword
plaintext password authentication failed
error code was NT_STATUS_NO_SUCH_USER (0xc0000064)
error message was: No such user
Could not authenticate user CHSchwartz%mypassword with
plaintext password
```

The error is absolutely normal in this case because there are no cleartext user credentials on the domain Controller (Active Directory) for this user.

```
challenge/response password authentication succeeded
[root@radiussrv1]#
```

As cleartext authentication fails, `wbinfo` tries a challenge/response. If a challenge/response succeeds, the Linux server is configured correctly to authenticate users against Active Directory!

Let's try to authenticate with NTLM, which is necessary for using FREERADIUS with Active Directory.

Type the following line:

```
[root@radiussrv1]# ntlm_auth –-request-nt-key
     –-domain=<your domain> –-username= <your username>
```

For me, the command would look like this:

```
[root@radiussrv1]# ntlm_auth –-request-nt-key
     –-domain=XYZDOM –-username= CHSchwartz
```

You will be prompted for your password.

The command line returns

```
NT_STATUS_OK : Success (0x0)
[root@radiussrv1]#
```

if the username and password are the same as those stored in Active Directory. Note that this mechanism is based on a challenge/response of the nt-key, a character string that has been encrypted with information taken from the username and password. During this operation, no exchange of user information takes place. Everything is based upon a comparison of encrypted strings.

IV Installation of FREERADIUS

Download first the latest source of Openssl (0.9.7f was used for my tests).

Extract the source files from tarball

```
~# tar –zxvf openssl-0.9.7f.tar
```

Install openssl in `/usr/local/openssl/`

```
~# ./config --prefix=/usr/local/openssl shared
~# make
~# make install
```

Download the latest version of FREERADIUS from www.freeradius.org.

This document refers to version 1.0.3 of FREERADIUS.

Install FREERADIUS with the following option:

```
~# ./configure –sysconfdir=/etc/

~# make
~# make install
```

In order to get FREERADIUS working, the following files must be configured:

- clients.conf
- radiusd.conf
- eap.conf
- users

IV.1 Configuration of clients.conf

Open the file `clients.conf` with your preferred editor.
It is located in /etc/raddb/.

Now we add a first Cisco switch which will be charged for access control.

Add:

```
# admswi3 cisco 3750

client 192.168.2.44 {
        secret          = 2!34r&dp0t
        shortname       = 192.168.2.44
        nastype         = cisco
}
```

The secret is a common password between the switch and RADIUS server. It is necessary to prevent the installation of wild access points. (Remember to set up the secret as well on your switch!)

Shortname is the IP address of your switch. (Watch out to use the correct one and not the IP address from this example!)

Nastype indicates the type of access point. In our case it is Cisco hardware.

Specify now the network(s) from which the access control is activated.

```
client 192.168.2.0/24 {
        secret          = 2!34r&dp0t
        shortname       = network1
}
```

Any computers that have an IP address outside this range will not be authorised for authentication. (Again, please use your networks addresses!)

IV.2 Configuration of radiusd.conf

Open this file and proceed to the section:
`# Microsoft CHAP authentication`

Make sure that the following lines are uncommented and that the value is the same as indicated here.

```
authtype = MS-CHAP

with_ntdomain_hack = yes
```

Ntdomain_hack is necessary to correct an error due to the challenge/response and the format in which the user information is sent.

The following line is the most important one. It allows using the Windows Domain Controller (Active Directory) for authentication.

```
ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key
           --domain=%{mschap:NT-Domain}
           --username=%{mschap:User-Name}
           --challenge=%{mschap:Challenge:-00}
           --nt-response=%{mschap:NT-Response:-00}"
```

IV.3 Configuration of eap.conf

Open the file eap.conf.

In this file we specify the authentication method used by FREERADIUS. We chose the PEAP (Protected EAP) method because it allows to use MSCHAPv2, a challenge/response protocol to authenticate against an Active Directory Windows Domain.

Replace the line « default_eap_type = md5 » with
« default_eap_type = peap » .

Proceed to section
`## EAP-TLS`

In order to get PEAP working, we need a TLS tunnel to encrypt communication between supplicant and RADIUS server. This means that we need server certificates. The production of self-signed server certificates is described in chapter VIII.

Uncomment the following lines:

```
tls {
        private_key_password = whatever
        private_key_file = ${raddbdir}/certs/cert-srv.pem

        certificate_file = ${raddbdir}/certs/cert-srv.pem

        #  Trusted Root CA list
        CA_file = ${raddbdir}/certs/demoCA/cacert.pem

        dh_file = ${raddbdir}/certs/dh
        random_file = ${raddbdir}/certs/random
        random_file = /dev/urandom
}
```

Replace the line `random_file = ${raddbdir}/certs/random` with `random_file = /dev/urandom`

Find and uncomment

```
peap {
           default_eap_type = mschapv2

     }
```

IV.4 Configuration of users

The configuration of this file is only necessary for advanced usage of FREERADIUS, like VLAN assignment or authentication of special users not included in AD and demanding other authentication methods.
This is not covered by this tutorial.

V Configuration of the switch

This configuration applies to the Cisco Catalyst 3750.
It can also be used for Catalysts 29xx.
Please read the software configuration guide of your switch fro any details!

Enter privileged EXEC mode of the switch.

Here are the commands to activate the switch for 802.1x port based authentication:

- Activate AAA (Authentication, Authorization, Accounting)

  ```
  #enable aaa new-model
  ```

- Create a list of authentication methods by using Radius group as default.

  ```
  #aaa authentication dot1x default group radius
  ```

- Activate authorization for using dynamic VLAN assignment by Radius.

  ```
  #aaa authorization network default group radius
  ```

- Configure parameters of Radius server. In this case we use IP 192.168.2.16 and the default ports 1812 et 1813

  ```
  # radius-server host 192.168.2.16 auth-port 1812
  acct-port 1813 timeout 3
  ```

- Configure the maximum number of retransmissions to the server for the requests (if there is no response of the server or if the server is slow).

  ```
  #radius server retransmit 3
  ```

- Configure the shared secret between switch and Radius server. Radius authentication can not work if the password does not match with the one of the Radius server.

  ```
  #radius server key <mysharedsecret>
  // please use the secret that you have specified in
     the clients.conf file (see chapter IV.1)
  ```

Next we must configure each interface (port) to operate in 802.1X mode.

Repeat this procedure for each port that should do access control:

```
#configure terminal
(config)#interface FastEthernet1/0/12
(config-if)# switchport mode access
(config-if)# dot1x port-control auto
(config-if)# end
```

The command `#show dot1x` allows to check 802.1x settings.

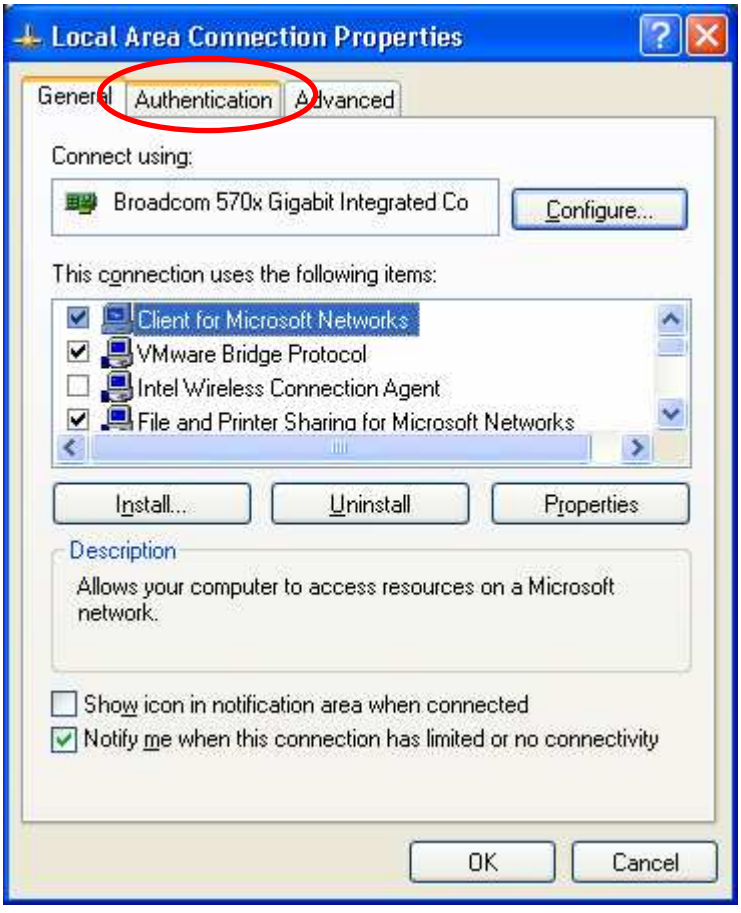
Do not forget to save the configuration of your switch.

```
#copy running-config startup-config
```

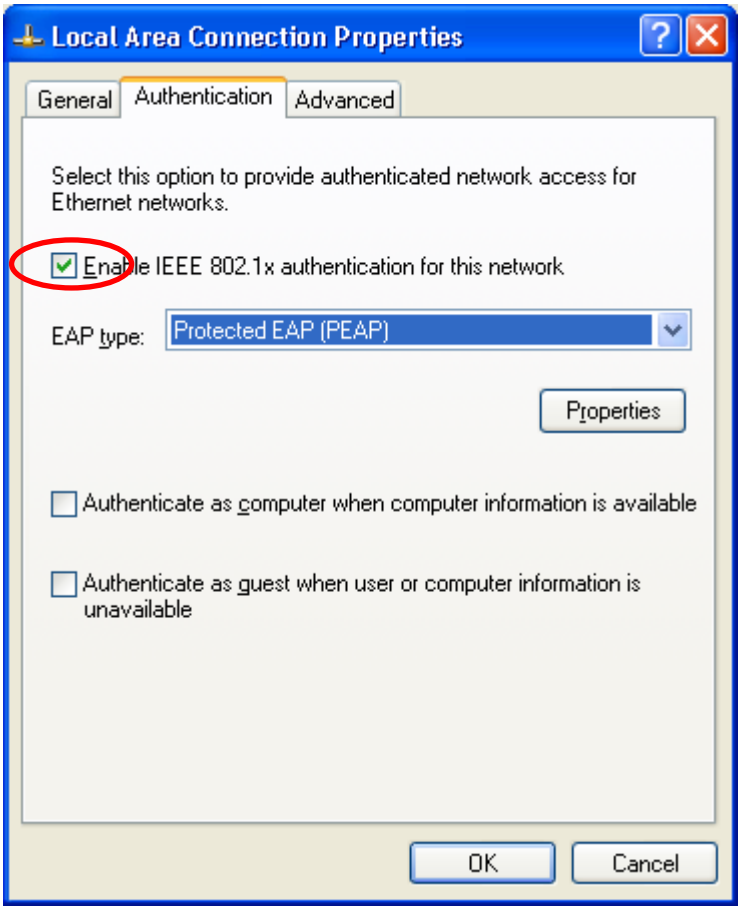
VI Configuration of the supplicant

This chapter illustrates the configuration of a Windows XP supplicant. No additional installation is needed to do this.
Unfortunately you cannot use this authentication method for Windows 2000 operating systems. Microsoft's patch (Q313664_W2K_SP4_X86_EN.exe) makes it only available for wireless interface adapters.
Perhaps third-party supplicant software may solve this issue, but I did not find any at the time of writing this tutorial.

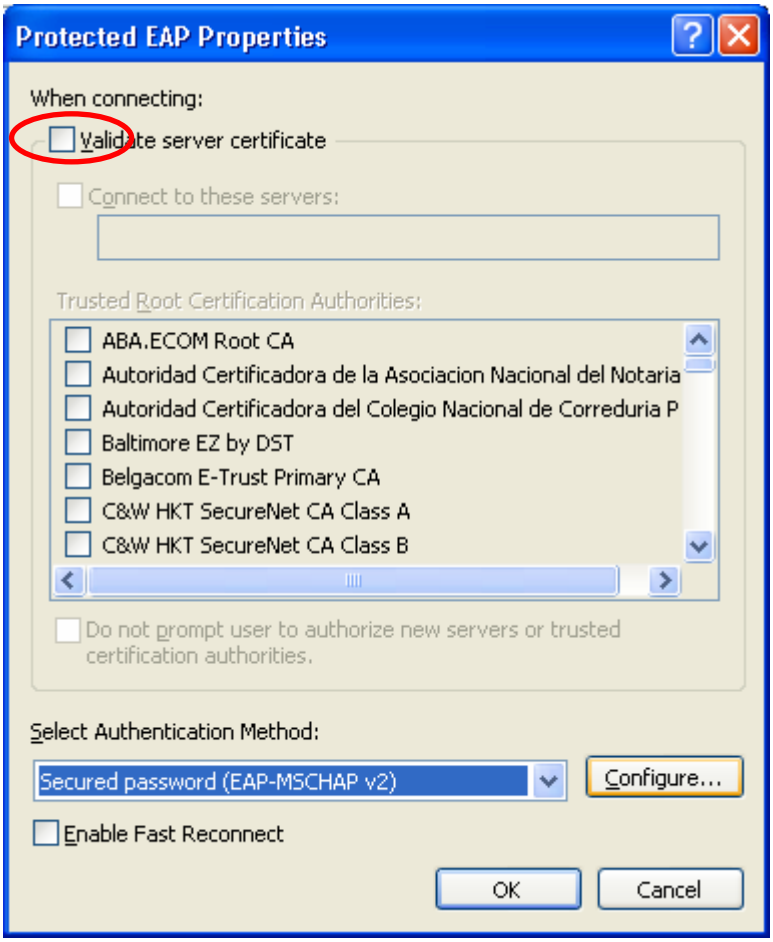
Open the network configuration panel select the network card and enter the properties.



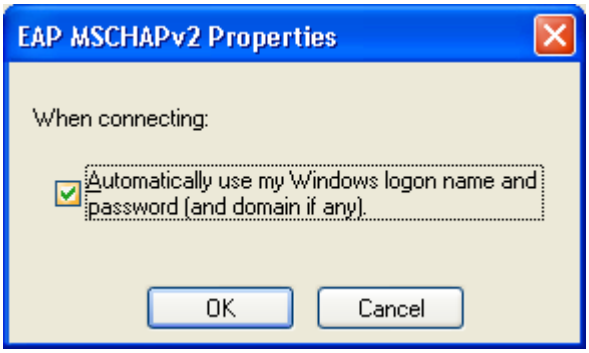
Select the « Authentication » tab.



Activate « Enable IEEE 802.1X authentication for this network »
For EAP type, chose « Protected EAP » from the list and then click the « Properties » button.



Deselect « Validate server certificate » and select « EAP-MSCHAP v2 » as authentication method. Click the « Configure » button next.



Check if « Automatically use my Windows logon name and password (and domain if any) » is activated. Otherwise the user will be prompted to authenticate after Windows login.

VII Self-signed certificates

TLS and PEAP require both server and client certificates. To generate the requested certificates, it is recommended to use the script « CA.all » that comes with FREERADIUS.
CA.all uses the configuration of the openssl.cnf file. It is possible to replace the certificates later by those obtained from a real certification authority.

Open the file openssl.cnf. It is located in /usr/local/openssl/ssl

Replace/add the following lines.
Note that the config file contains the password < whatever >. It's the certificate password.

Please replace the green items with the ones that correspond to your country and company

```
# req_extensions = v3_req

# The extensions to add to a certificate request

[ req_distinguished_name ]

countryName = Country Name (2 letter code)
countryName_default = LU
countryName_min = 2
countryName_max = 2

stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Luxembourg

localityName = Locality Name (eg, city)
localityName_default = Luxembourg-city

0.organizationName = Organization Name (eg, company)
0.organizationName_default = XYZ

organizationalUnitName = Organizational Unit Name
organizationalUnitName_default = IT

commonName = Common Name (eg, YOUR name)
commonName_max = 64
commonName_default = administrator

emailAddress = Email Address
emailAddress_max = 40
emailAddress_default = operations@xyz.com

# SET-ex3 = SET extension number 3
```

```
[ req_attributes ]

challengePassword = A challenge password
challengePassword_min = 4
challengePassword_max = 20
challengePassword_default = whatever

unstructuredName = An optional company name
```

We are going to use this information 3 times when executing CA.all.

During the first pass, this information produces the root certificates.
We can accept all default values while this pass.

The second pass produces the client certificates. We have to change the CommonName for the name of the client.

During the third pass, we have to change only the CommonName to the name of the server.

Before executing the script, check the following line in the CA.all script:

```
echo "newreq.pem" | /usr/local/openssl/ssl/misc/CA.pl –newca
```

CA.all is located in the scripts subdirectory of FREERADIUS' source directory.
In case of doubt try the command:

```
find / -name CA.all
```

When executing CA.all, we produce the 9 following certificates:

root.pem, root.p12, root.der
cert-clt.pem, cert-clt.p12, cert-clt.der
cert-srv.pem, cert-srv.p12, cert-srv.der

The server needs the files root.pem and cert.srv.pem in order to work with PEAP.

Move all the files to /etc/raddb/certs/. Do not forget the demoCA directory.

Good luck!