

## Installation et configuration : Auth Radius & Apache

ii	apache	1.3.31-7	Versatile, high-performance HTTP server
ii	apache-common	1.3.31-7	Support files for all Apache web servers
ii	apache-utils	1.3.31-7	Utility programs for web servers
ii	libapache-mod-radius	1.5.7-5	Apache module for RADIUS authentication

### ACTIVATION DU MODULE MOD\_AUTH\_CACHE A L'INSTALLATION

## /etc/apache/httpd.conf

```
##
## httpd.conf -- Apache HTTP server configuration file
##

ServerAdmin webmaster@reverse.exemple.com
ServerName reverse.exemple.com
<IfModule mod_auth_radius.c>
    AddRadiusAuth radius.exemple.com:1812 extranetweb 5:3
    AddRadiusCookieValid 20
</IfModule>
<IfModule mod_proxy.c>
    ProxyRequests Off
    <Directory proxy:*>
        AuthCache on
        AuthCacheSaveAuthorization on
        AuthCacheSuppressPort on
        AuthCacheSendFullPath 1
        AuthCacheDomainName exemple.com
        AuthCacheSendDomain 1
        AuthType basic
        AuthName "Authentification"
        AuthAuthoritative Off
        AuthRadiusAuthoritative On
        AuthRadiusActive On
        AuthRadiusCookieValid 20
        require valid-user
    </Directory>
    ProxyVia Off
</IfModule>
NameVirtualHost 193.4.186.89
Include /etc/apache/conf.d
```

## Radius Debian Sarge

Machine avec freeradius			
ii	freeradius	1.0.1-1	a high-performance and highly configurable R
ii	freeradius-mysql	1.0.1-1	MySQL module for FreeRADIUS server

## /etc/pam.d/radiusd

```
#
# /etc/pam.d/radiusd - PAM configuration for freeradius
#
# We fall back to the system default in /etc/pam.d/common-*
#
```

```
@include common-auth
#include common-account
#include common-password
#include common-session
```

## **/etc/freeradius/clients.conf**

```
#####
#
# clients.conf - client configuration directives
#
# This file is included by default.  To disable it, you will need
# to modify the CLIENTS CONFIGURATION section of "radiusd.conf".
#
#####

# Listes des machines ayant accès à l'authentification
client 192.168.0.25/32 {
    secret          = extranetweb
    shortname       = extranet-web-exemple
}

client 192.168.0.26/32 {
    secret          = extranetweb
    shortname       = extranet-web-exemple
}

client 192.168.25.15/32 {
    # secret and password are mapped through the "secrets" file.
    secret          = vpnexemple
    shortname       = serveur3
    # the following three fields are optional, but may be used by
    # checkrad.pl for simultaneous usage checks
}

```

## **/etc/freeradius/radiusd.conf**

```
authorize {
    #
    # The preprocess module takes care of sanitizing some bizarre
    # attributes in the request, and turning them into attributes
    # which are more standard.
    #
    # It takes care of processing the 'raddb/hints' and the
    # 'raddb/huntgroups' files.
    #
    # It also adds the %{Client-IP-Address} attribute to the request.
    preprocess
    #
    # If you want to have a log of authentication requests,
    # un-comment the following line, and the 'detail auth_log'
    # section, above.
    #
    # auth_log
    # attr_filter
    #
    # The chap module will set 'Auth-Type := CHAP' if we are
    # handling a CHAP request and Auth-Type has not already been set
    chap
    #
}

```

```

# If the users are logging in with an MS-CHAP-Challenge
# attribute for authentication, the mschap module will find
# the MS-CHAP-Challenge attribute, and add 'Auth-Type := MS-CHAP'
# to the request, which will cause the server to then use
# the mschap module for authentication.
mschap
#
# If you have a Cisco SIP server authenticating against
# FreeRADIUS, uncomment the following line, and the 'digest'
# line in the 'authenticate' section.
#
# digest
#
# Look for IPASS style 'realm/', and if not found, look for
# '@realm', and decide whether or not to proxy, based on
# that.
#
# IPASS
#
# If you are using multiple kinds of realms, you probably
# want to set "ignore_null = yes" for all of them.
# Otherwise, when the first style of realm doesn't match,
# the other styles won't be checked.
#
suffix
#
ntdomain
#
# This module takes care of EAP-MD5, EAP-TLS, and EAP-LEAP
# authentication.
#
# It also sets the EAP-Type attribute in the request
# attribute list to the EAP type from the packet.
eap

#
# Read the 'users' file
files
#
# 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
#
# The ldap module will set Auth-Type to LDAP if it has not
# already been set
#
# ldap
#
# Enforce daily limits on time spent logged in.
#
# daily
#
# Use the checkval module
#
# checkval
#
# Accounting. Log the accounting data.
#
accounting {
#
# Create a 'detail'ed log of the packets.
# Note that accounting requests which are proxied
# are also logged in the detail file.
#
# detail
#
# daily

```

```

# Update the wttmp file
#
# If you don't use "radlast", you can delete this line.
#
unix
#
# For Simultaneous-Use tracking.
#
# Due to packet losses in the network, the data here
# may be incorrect. There is little we can do about it.
radutmp
#
sradutmp
# Return an address to the IP Pool when we see a stop record.
#
main_pool
#
# Log traffic to an SQL database.
#
# See "Accounting queries" in sql.conf
sql

# Cisco VoIP specific bulk accounting
#
pgsql-voip
}

# 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 Querie" in sql.conf
    sql
}
# Post-Authentication
# Once we KNOW that the user has been authenticated, there are
# additional steps we can take.
post-auth {
    # Get an address from the IP Pool.
    #
    main_pool
    #
    # If you want to have a log of authentication replies,
    # un-comment the following line, and the 'detail reply_log'
    # section, above.
    #
    reply_log

#
# After authenticating the user, do another SQL qeury.
#
# See "Authentication Logging Queries" in sql.conf
sql

#
# Access-Reject packets are sent through the REJECT sub-section
# of the post-auth section.
#
Post-Auth-Type REJECT {
    #
    insert-module-name-here
    #
}

}

```

## /etc/freeradius/sql.conf

```
#
# Configuration for the SQL module, when using MySQL.
#
# The database schema is available at:
#
#   src/radiusd/src/modules/rlm_sql/drivers/rlm_sql_mysql/db_mysql.sql
#
# If you are using PostgreSQL, please use 'postgresql.conf', instead.
# If you are using Oracle, please use 'oracle.conf', instead.
# If you are using MS-SQL, please use 'mssql.conf', instead.
#
#   $Id: sql.conf,v 1.41.2.1 2004/06/10 00:45:01 phampson Exp $
#
sql {
    # Database type
    # Current supported are: rlm_sql_mysql, rlm_sql_postgresql,
    # rlm_sql_iodbc, rlm_sql_oracle, rlm_sql_unixodbc, rlm_sql_freetds
    driver = "rlm_sql_mysql"
    # Connect info
    server = "localhost"
    login = "admin"
    password = "motdepasseadmin"
    # Database table configuration
    radius_db = "radius"
}
```