AD/LDAP Authentication with OpenVMS

Installing the SYS$ACM (ACMELOGIN) Enabled LOGIN

To install the SYS$ACM enabled LOGIN and ACMELDAP kits, run the command file:

```
$ @SYS$MANAGER:SYS$LOGIN_SWITCH.COM
```

NOTE: This step only needs to be done once.

When the ACMELDAP agent is installed, you can setup the LDAP persona extension.

Setting Up the LDAP Persona Extension

To setup the persona extension, perform the following:

1. Install the persona extension image using the following commands:

```
$ MCR SYSMAN
SYSMAN> SYS_LOADABLE ADD LDAPACME LDAPACME$EXT
SYSMAN> exit
```

```
$ @SYS$UPDATE:VMS$SYSTEM_IMAGES.COM
```

2. Reboot the system:

```
$ @SYS$SYSTEM:SHUTDOWN
```

During the reboot, an error message may appear if the persona extension image is not loaded. If the error message is not displayed, it means the image loaded properly. After setting up the LDAP persona extension, you can move on to configuring the ACME LDAP agent.

Configuring the ACME LDAP Agent

Configuration of the ACME LDAP agent requires the following:

1. Editing the LDAP configuration file.
2. Starting the ACME LDAP agent.

Editing the LDAP Configuration File

To edit the ACME LDAP INI file, perform the following steps:

1. Make a copy of SYS$STARTUP:LDAPACME$CONFIG-STD.INI_TEMPLATE and rename using the file name of your choice. For example, use the following command:

```
$ COPY SYS$STARTUP:LDAPACME$CONFIG-STD.INI_TEMPLATE SYS$STARTUP:LDAPACME$CONFIG-
STD.INI
```

2. Edit SYS$STARTUP:LDAPACME$CONFIG-STD.INI to specify the directives that correspond to your AD/LDAP system.

Below is a sample INI file using HSQSVR.HSQ.COM as the AD/LDAP Host. Items highlighted in yellow are the edited fields from within the sample INI file using domain HSQ.COM as an example.

```
! Copyright 2013 Hewlett-Packard Development Company, L.P.
!
! This file is a template to help you create your own initialization
! file which will be read by the HP OpenVMS LDAP SYS$ACM Authentication Agent
! when it starts up.  It contains a set of information which determines
! how the agent should find the LDAP servers which contain authentication
! information.
!
! At start up time, the agent will use the logical name "LDAPACME$INIT"
! to find this file
!
! To create your own file, you can edit this file and replace the example
! parameters with information that corresponds with your own environment.
!
! Use the "server" directive to provide the IP address (or DNS host name)
! for your directory server.
!
! On OpenVMS V8.4 and above, you can specify one or more redundant servers
! by providing spaces between the server name/IP address.
!
! ex 1. server = test1.testdomain.com test2.testdomain.com
! ex 2. server = test1.testdomain.com test2.testdomain.com test3.testdomain.com
!
! The ACME LDAP tries to connect to first server first. If the connection fails
! for first server, the second server is tried for connection. If second server
! connection fails, the next set of server is tried in sequence, until the last
! server in the list. This applies to use search timeout as well.
!
! Note while using redundant servers:
!  1.)  The base_dn, bind_dn and bind_password should be same on all the redundant !
       directory servers. The user records getting authenticated using ACME LDAP
!      should also be present on all the directory servers.
!
!  2.)  Set the bind_timeout directive when using redundant multiple servers. This !
       ensures that the ACME LDAP tires to connect to all the redundant servers
!      before the user session times out.
!
```

```
!  3.)  In case you have provided the Certificate Authority's (CA) public key
!       (ca_file directive) and the public keys are different, provide all the
!       public keys in the same ca_file. See comments around ca_file directive.
!

server = 192.168.0.38

! Use the "port" tag to specify the LDAP port for connecting to the
! LDAP server. The default "port" is 389.

port = 389

! Use the "bind_dn" tag to specify an authentication distinguished name (DN)
! in LDAP format which the agent will use when binding to each of the
! servers in your list.

bind_dn = cn=administrator,cn=users,dc=hsq,dc=com

! Use the "bind_password" tag to specify a password to go with the
! authentication DN.
!

bind_password = XXXXXXXXXXXX

! Use the "bind_timeout" directive, if you are providing multiple redundant
! servers in the "server" directive.
!
! Each bind request to a directory server, will be default take around 75
! seconds (TCPIP default connection establishment timeout), if the directory
! server is not reachable.
!
! If there are multiple redundant servers, the user login session (say a
! telnet session) will expire (within approximately 30 seconds), before
! ACME LDAP agent could check the list of all servers mentioned in the
! "server" directive.
!
! The bind_timeout takes a timeout value in seconds for connecting to one
! directory server in the list of all servers mentioned in the "server" directive.
! If you have say 2 servers mentioned in the .server. directive and bind_timeout
! is set to 3 seconds, the overall timeout period is around 6 seconds.

bind_timeout = 3

!
! uncomment the following to use alternate server in case of search timeout while
! using redundant servers. The "server" directive can have more than one server
! mentioned as space (single) seperated list.
!
! search_timeout = 3
! Use the "base_dn" tag to specify the location in the directory underneath
! which the authentication records are kept:

base_dn = cn=users,dc=hsq,dc=com

! Attribute to map from principal to LDAP entry

login_attribute = samaccountname
```

```
! Scope to search for an LDAP entry
!  sub: searches the base entry and all entries at all levels below the base entry
!  one: searches all entries at one level below the base entry
!  base: searches only the base entry

scope = sub

! Filter for searching directory objects for valid user accounts (defaults
! to objectclass=*

! filter = objectclass=*

! Use the "port_security" directive to control how communications over the LDAP port
! are secured. The default is "starttls".
!
! The possible values for "port_security" are:
!
! starttls        (negotiate SSL/TLS over standard LDAP port)
! ssl             (this is an SSL-only port, e.g. port 636)
! none            (no security - not recommended)

port_security = none

! Password type for password changes
!  standard: use the standard userPassword attribute (default) on directory server
!  active-directory: use unicodePwd

password_type = active-directory

! Password update method for changes to the standard password attribute (userPassword)
!  standard:use the standard userPassword attribute (default) on directory server
!  active-directory:use unicodePwd

password_update = replace

! The LDAP SYS$ACM Authentication Agent will verify the validity of the LDAP
! server's public key certificate when using SSL. In order for this to happen
! you need to specify the location ("ca_file") of a file containing the Certificate
! Authority's (CA) public key used to sign the LDAP server's certificate.
!
! You can choose to disable this check by commenting out the line below.
!
! In case there are redundant servers having different public key certificate
! add the certificate information of the all the servers into the same file:
! example:
! $ type cacert.pem
! -----BEGIN CERTIFICATE-----
! .......
! server 1 public key certificate in base64 encoded format
! .......
! -----END CERTIFICATE-----
! -----BEGIN CERTIFICATE-----
! .......
! server 2 public key certificate in base64 encoded format
! .......
! -----END CERTIFICATE-----
! $
```

```
! ca_file = [directory]cacert.pem

!
! mapping for user name mapping whether global or local
!   Possible options are:
!
!   mapping is commented:
!            If mapping is commented, one-to-one mapping is used.
!            i.e. user name at "username: " prompt is the same as in sysuaf.dat file
!
!   server:  (global mapping) Mapping between user name entered at "Username:" prompt
!            and the sysuaf.dat user account name happens based on some attributes
!            on the directory server.
!            You need to provide the mapping_attribute and mapping_target directive
!            if you use mapping=server
!
!    local:  Mapping between user name entered at "Username:" prompt and the sysuaf.dat
!            user account name happens based on local CSV database file.
!            You need to provide the mapping_file directive if you use mapping=local

! mapping = local

! This directive is applicable only for global mapping.
! Specifies the attribute on directory server that is used for user mapping.
! For example:
! mapping_attribute can be referenced to the description attribute for the user
! in the directory server.
! For example: Let the LDAP INI file have :
!    mapping_attribute=description
!    mapping_target=VMSUsers.hp.com
! Let the description (attribute in Directory Server) be populated with:
!   VMSUsers,hp.com/jdoe
! The LDAP ACME agent then searches in VMSUsers.hp.com/jdoe, for a prefix of
! VMSUsers.hp.com/ (with a forward slash (/) along with the mapping_target).
! The rest of the value that is, jdoe. is considered as the user name present in
! SYSUAF.DAT file. If a multi-valued string attribute is used, the VMSUsers.hp.com/jdoe
! must be one of the array elements of the multi-valued string.

! mapping_target=VMSUsers.hp.com

! This directive is applicable only for local mapping.
!
! Specifies the complete path of the text database file to be searched for mapping users.
! A template file is available in SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT_TEMPLATE.
! This file includes the LDAP username and VMS usernames separated by a comma, where LDAP
! username is the name of the user in the domain.
!
! For information on how to populate and load the contents of the database file, see
! SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT_TEMPLATE

! This directive is applicable only for Multi-Domain support
!
! Specifies the domain name of the ldap directory server against which the users has
to be authenticated.
!
!domain = testdomain1
```

Editing the ACME Start File

Edit SYS$MANAGER:ACME$START.COM and define the following logical name:

```
$ DEFINE/SYSTEM/EXECUTIVE LDAPACME$INIT SYS$STARTUP:LDAPACME$CONFIG-STD.INI
```

The LDAPACME$INIT logical must contain the path name to the initialization for the ACME LDAP Agent Server.

Remove the comment from the following line from SYS$MANAGER:ACME$START.COM:

```
$! @SYS$STARTUP:LDAPACME$STARTUP-STD ! LDAP
```

Ensure that the LDAP configuration file and the LDAP local database mapping file are accessible for privileged users only. You can set the security of these files appropriately, based on your security requirements. For example, the following command sets the accessibility of LDAPACME$CONFIG-STD.INI and LDAP_LOCALUSER_DATABASE.TXT files only for the system user:

```
SET SECURITY / PROTECTION = (system:"RWED", OWNER:"", GROUP:"", WORLD:"")
        SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD.INI
SET SECURITY / PROTECTION = (system:"RWED", OWNER:"", GROUP:"", WORLD:"")
        SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE.TXT
```

Starting the ACME LDAP Agent

To restart the ACME_SERVER process:

```
$ SET SERVER ACME/EXIT/WAIT
$ SET SERVER ACME/START=AUTO
```

Specifying EXTAUTH and VMSAUTH Flags on OpenVMS

For users to be externally authenticated (via LDAP), the ExtAuth flag has to be set for the user account in SYSAUF.DAT. When the ExtAuth flag is specified for a user account, the user is validated only using the external authenticator (LDAP). If you want this user to be authenticated locally as well, set the VMSAuth flag for the user account in the SYSAUF.DAT file and use the "/local" qualifier during login.

To set the ExtAuth flag for the user, enter the following:

```
$ SET DEFAULT SYS$SYSTEM
$ MCR AUTHORIZE MODIFY <username> /FLAGS=(EXTAUTH,VMSAUTH)
```