LDAP Authentication for 3PAR InServs

A Configuration Guide

As 3PAR storage arrays appear in larger mainstream datacenters, we find more customers wanting to integrate with their Active Directory infrastructure for user authentication and authorization. InForm OS can use Active Directory authentication if you configure it to use the Lightweight Directory Access Protocol (LDAP) for authentication.

The *HP InForm OS 3.1.1 CLI Administrator's Manual* addresses the topic of LDAP authentication in some detail but it leaves rather a lot unsaid. This document is intended to be a brief practical guide to getting Active Directory authentication configured and working. Although 3PAR arrays can work with other LDAP implementations, this document will limit itself to configuring authentication with Microsoft Active Directory.

Authentication and authorization are two related but independent concepts. *Authentication* is the process of determining if a user is recognized as a valid user. Anyone who presents recognized credentials (e.g., an active user name and its correct password) is considered a valid user and is authenticated. *Authorization* goes hand-in-glove with authentication to determine the privileges that are associated with the user. An authenticated user may be granted access, but the authentication process determines what, if anything, that user is allowed to do. It is possible that a user may be authenticated not have any authorized capabilities.

The following steps walk you through configuring LDAP authentication. You must use the command line interface. The InForm Management Console (IMC) does not support LDAP configuration.

1. Clear any previous authentication configuration

   The simplest way to clear a previous authentication configuration is to use the *setauthparam* command as shown:

   ```
   cli% setauthparam –f –clearall
   ```

   The *–clearall* flag clears all current authentication configuration parameters. Including *–f* on the command line suppresses the confirmation prompt. We will use this on all of our subsequent *setauthparam* commands.

   If you believe the InServ does not have authentication configured, use the *showauthparam* command to confirm this:

   ```
   cli% showauthparam
   Param -Value-
   ```

   If *showauthparam* prints only column headers and nothing else, no authentication parameters have been configured and it is not necessary to clear them.

2. Configure the parameters required to connect to the Active Directory server

   The first set of parameters identify the Active Directory server by name and IP address and specify the Kerberos realm that will allow the InServ to query AD. Substitute values appropriate for your customer's environment for the parameter values shown in this example:

   ```
   cli% setauthparam -f ldap-server 192.168.10.33
   cli% setauthparam -f ldap-server-hn larry.stooges.com
   cli% setauthparam -f kerberos-realm STOOGES.COM
   ```

   `ldap-server`
   : Specify the IP address of your customer's Active Directory server.

   `ldap-server-hn`
   : Specify the host name of the Active Directory server. DNS must be able to look up this host name and return the IP address specified for *ldap-server* and vice versa.

   `kerberos-realm`
   : Specify the LDAP service name. Typically this is the same as the DNS domain name but check the AD configuration to be sure.

3. Configure the binding parameters

   The next two commands configure the way the InServ will bind to the Active Directory server. Enter these two commands as shown without substitutions.

   ```
   cli% setauthparam -f binding sasl
   cli% setauthparam -f sasl-mechanism GSSAPI
   ```

   `binding`
   : This parameter specifies the binding type, in this case, SASL, which stands for Simple Authentication and Security Layer.

   `sasl-mechanism`
   : Specifies GSSAPI (Generic Security Services Application Program Interface) as the SASL binding mechanism. Other valid values are PLAIN and DIGEST-MD5, but use GSSAPI.

4. Configure the account location parameters

   Account location parameters specify where in the Active Directory looks in its tree structure to find valid users and how to look for them.

   ```
   cli% setauthparam -f account-obj user
   cli% setauthparam -f account-name-attr sAMAccountName
   cli% setauthparam -f memberof-attr memberOf
   cli% setauthparam -f accounts-dn "OU=Domain Admins,DC=Stooges,DC=com"
   ```

   `account-obj`
   `account-name-attr`
   `memberof-attr`
   : Specify each of these exactly as shown. Together they specify an AD search that will find a user that is a member of a specified group.

`accounts-dn`

This specifies the group that contains the user. The double quotes are only necessary if the search string contains spaces.

Any user that is a member of this group (or its subordinate groups) will be authenticated as a valid user. Use the Windows Active Directory Users and Computers graphical interface or a tool such as Sysinternals' ADExplorer to help you determine with this search string should be.

Note that an authenticated user has no privileges until it is authorized as well.

5. Configure authorization

Specify authorization levels by configuring one or more "map" parameters. A map parameter identifies an AD group whose members are authorized for specific roles within the InServ. For example, any authenticated AD user who is a member of the group specified by the *super-map* parameter will be authorized for the *super* role. InServ OS recognizes the following map parameters:

Map Name	**Authorized Capabilities**
`3PAR_AO-map`	3PAR Adaptive Optimization rights
`3PAR_RM-map`	3PAR Recovery Manager rights
`basic_edit-map`	Basic editing
`browse-map`	Browse (read only)
`create-map`	Create objects
`edit-map`	Edit objects
`service-map`	Service rights
`super-map`	Super user rights (all except Service)

Map parameters are specified as follows:

```
cli% setauthparam -f super-map "CN=3PAR,OU=Security Groups,DC=Stooges,DC=com"
cli% setauthparam -f browse-map "CN=Users,OU=Security Groups,DC=Stooges,DC=com"
```

`super-map`

A valid (authorized) user who is a member of the *3PAR* group will be granted the InForm OS super user role.

`browse-map`

A valid user who is a member of the *Users* group will be able to look at InServ objects (browse) but will not be able to change anything.

6. Review the configuration

Use the *showauthparam* command to print the authorization parameters and review them:

```
%cli showauthparam
Param              --------------------------Value--------------------------
ldap-server        192.168.10.33
ldap-server-hn     larry.stooges.com
kerberos-realm     STOOGES.COM
binding            sasl
sasl-mechanism     GSSAPI
accounts-dn        OU=Domain Admins,DC=Stooges,DC=com
account-obj        user
account-name-attr  sAMAccountName
```

```
memberof-attr     memberOf
super-map         CN=Admins,OU=Security Groups,DC=Stooges,DC=com
browse-map        CN=Users,OU=Security Groups,DC=Stooges,DC=com
```

The display should accurately reflect the parameters you entered. Carefully check the *accounts-dn* and map parameters to ensure that you typed the search strings correctly.

7. Test authorization

You can use the *checkpassword* command to test your LDAP configuration. *Checkpassword* attempts to authenticate a user name first as a local user and then using LDAP. Supply the name of any AD user and enter the user's password when prompted. If you do not specify a user ID, the *checkpassword* command will attempt to validate the currently logged in user.

```
cli% checkpassword moe

password:
+ attempting authentication and authorization using system-local data
+ authentication denied: unknown username
+ attempting authentication and authorization using LDAP
+ using Kerberos configuration file:
[domain_realm]
larry.stooges.com = STOOGES.COM
[realms]
STOOGES.COM = {
kdc = larry.stooges.com
}
+ temporarily setting name-to-address mapping: larry.stooges.com ->
       192.168.10.33
+ attempting to obtain credentials for "moe@STOOGES.COM"
+ authentication/authorization denied: timeout waiting for server response
+ connecting to LDAP server using URI: ldap://larry.stooges.com
+ binding to user "moe" with SASL mechanism GSSAPI
+ searching LDAP using:
search base:    OU=Domain Admins,DC=Stooges,DC=com
scope:          sub
filter:         (&(objectClass=user)(sAMAccountName=moe))
for attributes: memberOf
+ search result DN: CN=Moe Howard,OU=Domain Admins,DC=stooges,DC=com
+ search result:    memberOf: CN=3PAR,OU=Security Groups,DC=stooges,DC=com
+ search result:    memberOf: CN=OA Admins,OU=Security Groups,DC=stooges,DC=com
+ search result:    memberOf: CN=CSAdministrator,CN=Users,DC=stooges,DC=com
+ search result:    memberOf: CN=Recipient Management,OU=Microsoft Exchange
       Security Groups,DC=stooges,DC=com
+ search result:    memberOf: CN=Organization Management,OU=Microsoft Exchange
       Security Groups,DC=stooges,DC=com
+ search result:    memberOf: CN=Domain Admins,CN=Users,DC=stooges,DC=com
+ search result:    memberOf: CN=Schema Admins,CN=Users,DC=stooges,DC=com
+ mapping rule: super mapped to by "CN=3PAR,OU=Security
       Groups,DC=Stooges,DC=com"
+ rule match: super mapped to by "CN=3par,OU=Security Groups,DC=stooges,DC=com"
user moe is authenticated and authorized
```

Command output shows the progress of the search through active directory. The last line indicates that the user (*moe* in this case) is both authenticated as a valid user and authorized with InForm OS "super" privileges.

References

HP 3PAR InForm OS 3.1.1 CLI Administrator's Manual

HP 3PAR InForm OS 3.1.1 Command Line Interface Reference

HP 3PAR InForm OS 3.1.1 Concepts Guide