LDAP on IOS Devices Using Dynamic Attribute Maps Configuration Example

TAC **Document ID: 113689**

Contributed by Atri Basu, Shaik Zubair, and Craig Lorentzen, Cisco
TAC Engineers.

Jan 17, 2013

Contents

Introduction

This document describes how to use Lightweight Directory Access Protocol (LDAP) authentication on Cisco IOS® headends and change the default Relative Distinguished Name (RDN) from Common Name (CN) to sAMAccountName.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on a Cisco IOS device that runs Cisco IOS Software Release 15.0 or later.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Core Issue

Most Microsoft Active Directory (AD) with LDAP users typically define their RDN to be the sAMAccountName. If you use authentication proxy (auth−proxy) and an Adaptive Security Appliance (ASA) as a headend for your VPN clients, this is easily fixed if you define the AD server type when you define the AAA server or if you enter the **ldap−naming−attribute** command. However, in the Cisco IOS software, neither of these options is available. By default, the Cisco IOS software uses the CN attribute value in AD for username authentication. For example, a user is created in AD as *John Fernandes*, but his user ID is stored as *jfern*. By default, the Cisco IOS software checks the CN value. That is, the software checks *John Fernandes* for username authentication and not the sAMAccountName value of *jfern* for authentication. In order to force the Cisco IOS software to check the username from the sAMAccountName attribute value, use dynamic attribute maps as detailed in this document.

Solution

Although Cisco IOS devices do not support these methods of RDN modification, you can use dynamic attribute maps in the Cisco IOS software in order to achieve a similar result. If you enter the **show ldap attribute** command on the Cisco IOS headend, you will see this output:

LDAP Attribute	Format	AAA Attribute
airespaceBwDataBurstContract	Ulong	bsn− data−bandwidth−burst−contr
userPassword	String	password
airespaceBwRealBurstContract	Ulong	bsn−realtime−bandwidth−burst−c
employeeType	String	employee−type
airespaceServiceType	Ulong	service−type
airespaceACLName	String	bsn−acl−name
priv−lvl	Ulong	priv−lvl
memberOf	String DN	supplicant−group
cn	**String**	**username**
airespaceDSCP	Ulong	bsn−dscp
policyTag	String	tag−name
airespaceQOSLevel	Ulong	bsn−qos−level
airespace8021PType	Ulong	bsn−8021p−type
airespaceBwRealAveContract	Ulong	bsn−realtime−bandwidth−average
airespaceVlanInterfaceName	String	bsn−vlan−interface−name
airespaceVapId	Ulong	bsn−wlan−id
airespaceBwDataAveContract	Ulong	bsn−data−bandwidth−average−con
sAMAccountName	String	sam−account−name
meetingContactInfo	String	contact−info
telephoneNumber	String	telephone−number

As you can see from the attribute highlighted, the Cisco IOS Network Access Device (NAD) uses this attribute map for authentication requests and for responses. Basically, a dynamic LDAP attribute map in the Cisco IOS device functions bidirectionally. In other words, attributes are mapped not only when a response is received, but also when LDAP requests are sent out. Without any user−defined attribute maps, a basic LDAP configuration on the NAD, you see this log message when the request is sent out:

```
*Jul 24 11:04:50.568: LDAP: Check the default map for aaa type=username
*Jul 24 11:04:50.568: LDAP: Ldap Search Req sent
ld 1054176200
base dn DC=csco,DC=com
scope 2
filter (&(objectclass=*)(cn=xyz))ldap_req_encode
put_filter "(&(objectclass=person)(cn=xyz))"
put_filter: AND
put_filter_list "(objectclass=person)(cn=xyz)"
put_filter "(objectclass=person)"
put_filter: simple
put_filter "(cn=xyz)"
put_filter: simple
Doing socket write
*Jul 24 11:04:50.568: LDAP: LDAP search request sent successfully (reqid:13)
```

In order to change this behavior and force it to use the sAMAccountName attribute for username verification, enter the **ldap attribute map username** command to create this dynamic attribute map first:

```
ldap attribute map username
   map type sAMAccountName username
```

Once this attribute map has been defined, enter the **attribute map** *<dynamic−attribute−map−name>* command to map this attribute map to the selected AAA server group (aaa−server).

Note: In order to make this entire process easier, the Cisco bug ID CSCtr45874 (registered customers only) has been filed. If this enhancement request is implemented, it will allow users to identify what kind of LDAP server is being used and automatically change some of these default maps to reflect the values used by that particular server.

Configure

In this section, you are presented with the information to configure the features described in this document.

Note: Use the Command Lookup Tool (registered customers only) to obtain more information on the commands used in this section.

Sample Configuration

This document uses these configurations:

- Enter this command in order to define the dynamic attribute map:

  ```
  ldap attribute map <dynamic−attribute−map−name>


  map type sAMAccountName username
  ```
- Enter this command in order to define the AAA server group:

```
aaa group server ldap <server-group-name>

server <server-name>
```

- Enter this command in order to define the server:

```
ldap server <server-name>

   ipv4 <host-address>

attribute map <dynamic-attribute-map-name>

bind authentication root-dn <complete-dn-root-user> password <root-user-pwd>

base-dn <complete-dn-search-base>
```

- Enter this command in order to define the list of authentication methods to use:

```
aaa authentication login <name> group <server-group-name>
```

AD Tools

In order to check the absolute Distinguised Name (DN) of a user, enter one of these commands from the AD command prompt:

```
dsquery user -name user1
```

OR

```
dsquery user -samid user1
```

Note: "user1" mentioned above is in regex string. You can also enlist all DNs of username starting with user by using the regex string as "user*".

In order to enlist all the attributes of a single user, enter this command from the AD command prompt:

```
dsquery * -filter "(&(objectCategory=Person)(sAMAccountName=username))" -attr *
```

Potential Problems

In an LDAP deployment, the search operation is performed first, and the bind operation is performed later. This operation is performed because, if the password attribute is returned as part of the search operation, the password verification can be done locally on the LDAP client and there is no need for an extra bind operation. If the password attribute is not returned, a bind operation can be performed later. Another advantage when you perform the search operation first and the bind operation later is that the DN received in the search result can be used as the user DN instead of the formation of a DN when the username (CN value) is prefixed with a base DN.

There might be issues when the **authentication bind−first** command is used along with a user−defined attribute which changes where the username attribute map points. For example, if you use this configuration, you are likely to see a failure in your authentication attempt:

```
ldap server ss−ldap
ipv4 192.168.1.3
attribute map ad−map
transport port 3268
bind authenticate root-dn CN=abcd,OU=Employees,OU=qwrt Users,DC=qwrt,DC=com
   password blabla
base−dn DC=qwrt,DC=com
authentication bind−first
ldap attribute−map ad−map
 map type sAMAccountName username
```

As a result, you will see the `Invalid credentials, Result code =49` error message. The log messages will look similar to these:

```
Oct  4 13:03:08.503: LDAP: LDAP: Queuing AAA request 0 for processing
Oct  4 13:03:08.503: LDAP: Received queue event, new AAA request
Oct  4 13:03:08.503: LDAP: LDAP authentication request
Oct  4 13:03:08.503: LDAP: Attempting first  next available LDAP server
Oct  4 13:03:08.503: LDAP: Got next LDAP server :ss−ldap
Oct  4 13:03:08.503: LDAP: First Task: Send bind req
Oct  4 13:03:08.503: LDAP: Authentication policy: bind−first
Oct  4 13:03:08.503: LDAP: Dynamic map configured
Oct  4 13:03:08.503: LDAP: Dynamic map found for aaa type=username
Oct  4 13:03:08.503: LDAP: Bind: User−DN=sAMAccountName=abcd,DC=qwrt,DC=com
ldap_req_encode
Doing socket write
Oct  4 13:03:08.503: LDAP:  LDAP bind request sent successfully (reqid=36)
Oct  4 13:03:08.503: LDAP: Sent the LDAP request to server
Oct  4 13:03:08.951: LDAP: Received socket event
Oct  4 13:03:08.951: LDAP: Checking the conn status
Oct  4 13:03:08.951: LDAP: Socket read event socket=0
Oct  4 13:03:08.951: LDAP: Found socket ctx
Oct  4 13:03:08.951: LDAP: Receive event: read=1, errno=9 (Bad file number)
Oct  4 13:03:08.951: LDAP: Passing the client ctx=314BA6ECldap_result
wait4msg (timeout 0 sec, 1 usec)
ldap_select_fd_wait (select)
ldap_read_activity lc 0x296EA104
Doing socket read
LDAP−TCP:Bytes read = 109
ldap_match_request succeeded for msgid 36 h 0
changing lr 0x300519E0 to COMPLETE as no continuations
removing request 0x300519E0 from list as lm 0x296C5170 all 0
ldap_msgfree
ldap_msgfree
Oct  4 13:03:08.951: LDAP:LDAP Messages to be processed: 1
Oct  4 13:03:08.951: LDAP: LDAP Message type: 97
Oct  4 13:03:08.951: LDAP: Got ldap transaction context from reqid
   36ldap_parse_result
Oct  4 13:03:08.951: LDAP: resultCode:    49     (Invalid credentials)
Oct  4 13:03:08.951: LDAP: Received Bind Responseldap_parse_result
   ldap_err2string
Oct  4 13:03:08.951: LDAP: Ldap Result Msg: FAILED:Invalid credentials,
   Result code =49
Oct  4 13:03:08.951: LDAP: LDAP Bind operation result : failed
Oct  4 13:03:08.951: LDAP: Restoring root bind status of the connection
Oct  4 13:03:08.951: LDAP: Performing Root−Dn bind operationldap_req_encode
Doing socket write
```

```
Oct  4 13:03:08.951: LDAP: Root Bind on CN=abcd,DC=qwrt,DC=com
initiated.ldap_msgfree
Oct  4 13:03:08.951: LDAP: Closing transaction and reporting error to AAA
Oct  4 13:03:08.951: LDAP: Transaction context removed from list [ldap reqid=36]
Oct  4 13:03:08.951: LDAP: Notifying AAA: REQUEST FAILED
Oct  4 13:03:08.951: LDAP: Received socket event
Oct  4 13:03:09.491: LDAP: Received socket event
Oct  4 13:03:09.491: LDAP: Checking the conn status
Oct  4 13:03:09.491: LDAP: Socket read event socket=0
Oct  4 13:03:09.491: LDAP: Found socket ctx
Oct  4 13:03:09.495: LDAP: Receive event: read=1, errno=9 (Bad file number)
Oct  4 13:03:09.495: LDAP: Passing the client ctx=314BA6ECldap_result
wait4msg (timeout 0 sec, 1 usec)
ldap_select_fd_wait (select)
ldap_read_activity lc 0x296EA104
Doing socket read
LDAP-TCP:Bytes read= 22
ldap_match_request succeeded for msgid 37 h 0
changing lr 0x300519E0 to COMPLETE as no continuations
removing request 0x300519E0 from list as lm 0x296C5170 all 0
ldap_msgfree
ldap_msgfree
Oct  4 13:03:09.495: LDAP: LDAP Messages to be processed: 1
Oct  4 13:03:09.495: LDAP: LDAP Message type: 97
Oct  4 13:03:09.495: LDAP: Got ldap transaction context from reqid
   37ldap_parse_result
Oct  4 13:03:09.495: LDAP: resultCode:    0     (Success)P: Received Bind
   Response
Oct  4 13:03:09.495: LDAP: Received Root Bind Response ldap_parse_result
Oct  4 13:03:09.495: LDAP: Ldap Result Msg: SUCCESS, Result code =0
Oct  4 13:03:09.495: LDAP: Root DN bind Successful on:CN=abcd,DC=qwrt,DC=com
Oct  4 13:03:09.495: LDAP: Transaction context removed from list [ldap reqid=37]
ldap_msgfree
ldap_result
wait4msg (timeout 0 sec, 1 usec)
ldap_select_fd_wait (select)
ldap_err2string
Oct  4 13:03:09.495: LDAP: Finished processing ldap msg, Result:Success
Oct  4 13:03:09.495: LDAP: Received socket event
```

The highlighted lines indicate what is wrong with the initial bind before authentication. It will work properly if you remove the **authentication bind−first** command from the above configuration.

Verify

Use this section to confirm that your configuration works properly.

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

- **show ldap attributes**
- **show ldap server all**

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

Troubleshooting Commands

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

Note: Refer to Important Information on Debug Commands before you use **debug** commands.

- **debug ldap all**
- **debug ldap event**
- **debug aaa authentication**
- **debug aaa authorization**

Related Information

- **AAA LDAP Configuration Guide Cisco IOS Release 15.1MT**
- **ASA 8.0: Configure LDAP Authentication for WebVPN Users**
- **Technical Support & Documentation − Cisco Systems**

Contacts & Feedback | Help | Site Map

Updated: Jan 17, 2013 Document ID: 113689