Introduction

This document describes how to configure OpenLDAP with custom schema to support per-user attributes for Cisco Anyconnect Secure Mobility Client that connects to a Cisco Adaptive Security Appliance (ASA). The ASA configuration is quite basic as all user attributes are retrieved from the OpenLDAP server. Also described in this document are the differences in LDAP authentication and authorization when used along with certificates.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic knowledge about Linux configuration
- Basic knowledge about ASA CLI configuration

Components Used

The information in this document is based on these software versions:

- Cisco ASA version 8.4 and later
- OpenLDAP version 2.4.30
Configure

Basic OpenLDAP Configuration

Step 1. Configure the server.

This example uses test-cisco.com ldap tree.

ldap.conf file is used to set system level defaults that can be used by local ldap client.

Note: Although you are not required to set up system-level defaults, they can help test and troubleshoot the server when you run a local ldap client.

/etc/openldap/ldap.conf:

slapd.conf file is used for OpenLDAP server configuration. Default schema files include widely used LDAP definitions. For example, the object class name person is defined in the core.schema file. This configuration uses that common schema and define its own schema for Cisco-specific attributes.

/etc/openldap/slapd.conf:

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema

# Defines backend database type and redirects all # queries with specified suffix to that
database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30

# Rootdn will be used to perform all administrative tasks.
rootdn "cn=Manager,dc=test-cisco,dc=com"

# Cleartext passwords, especially for the rootdn, should be avoid.
rootpw secret
directory /var/lib/openldap-data
index objectClass eq
```

Step 2. Verify the LDAP configuration.

In order to verify that basic OpenLDAP works, run this configuration:

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema

# Defines backend database type and redirects all # queries with specified suffix to that
database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30
```
# Rootdn will be used to perform all administrative tasks.

```bash
rootdn "cn=Manager,dc=test-cisco,dc=com"
```

# Cleartext passwords, especially for the rootdn, should be avoid.

```bash
rootpw secret
directory /var/lib/openldap-data
index objectClass eq
```

**Step 3. Add records to the database.**

Once you hve tested and configured everthing properly, add records to the database. In order to add basic containers for users and groups, run this configuration:

```bash
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema
```

This defines backend database type and redirects all # queries with specified suffix to that database.

```bash
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30
```

# Rootdn will be used to perform all administrative tasks.

```bash
rootdn "cn=Manager,dc=test-cisco,dc=com"
```

# Cleartext passwords, especially for the rootdn, should be avoid.

```bash
rootpw secret
directory /var/lib/openldap-data
index objectClass eq
```

**Custom Openldap Schema**

Now that the basic configuration works, you can add custom schema. In this configuration example, a new type of object class named *CiscoPerson* is created and these attributes are created and used in this object class:

- CiscoBanner
- CiscoACLin
- CiscoDomain
- CiscoDNS
- CiscoIPAddress
- CiscoIPNetmask
- CiscoSplitACL
- CiscoSplitTunnelPolicy
- CiscoGroupPolicy

**Step 1. Create the new schema in cisco.schema.**

```bash
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema
```
# Defines backend database type and redirects all # queries with specified suffix to that database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30

# Rootdn will be used to perform all administrative tasks.
rootdn "cn=Manager,dc=test-cisco,dc=com"

# Cleartext passwords, especially for the rootdn, should be avoid.
rootpw secret
directory /var/lib/openldap-data
index objectClass eq

**Important Notes**

- Use private enterprise OIDs for your company. Any OIDs will work, but best practice is to use the OIDs assigned by IANA. The one configured in this examples begins from 1.3.6.1.4.1.9 (which is reserved by Cisco: [http://www.iana.org/assignments/enterprise-numbers](http://www.iana.org/assignments/enterprise-numbers)).
- The following part of OID (500.1.1-500.1.9) has been used to not interfere directly in the main tree of the Cisco OID (“1.3.6.1.4.1.9”).
- This database uses the *Person* object class defined in schema/core.ldif. That object is of TOP type and records can include only one such attribute (which is why the *CiscoPerson* object class is of Auxiliary type).
- The object class named *CiscoPerson* must include SN or CN and can include any of the custom Cisco attributes defined earlier. Note that it can also include any other attributes defined in other schemas (such as *userPassword* or *telephoneNumber*).
- Remember that each object should have a different OID number.
- Custom attributes are case insensitive and of *string* type with UTF-8 encoding and maximum 128 characters (defined by SYNTAX).

**Step 2. Include the schema in slapd.conf.**

```bash
pluton openldap # cat slapd.conf | grep include
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/cisco.schema
```

**Step 3. Restart services.**

```bash
pluton openldap # cat slapd.conf | grep include
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/cisco.schema
```

**Step 4. Add a new user with all custom attributes.**

In this example, the user belongs to multiple objectClass objects, and it inherits attributes from all of them. With this process it is easy to add additional schema or attributes without changes to existing database records.

```bash
pluton # cat users.ldiff
```
Step 5. Set the password for the user.

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: (CRYPT)*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1
pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

**Step 6. Verify the configuration.**

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: (CRYPT)*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoAACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

**ASA Configuration**

**Step 1. Configure the interface and certificate.**

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

Step 2. Generate a self-signed certificate.

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: (CRYPT)*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

Step 3. Enable WebVPN on the outside interface.

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: (CRYPT)*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

Step 4. Split the ACL configuration.

The ACL name is returned by OpenLDAP:

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: (CRYPT)*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

Step 5. Create a tunnel-group name that uses the default group-policy (DfltAccessPolicy).

Users with the specific LDAP attribute (CiscoGroupPolicy) are mapped to another policy: POLICY1
ASA aaa-server configuration uses ldap attribute-map for mapping from attributes returned by OpenLDAP to attributes that can be interpreted by ASA for Anyconnect users.
pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

Step 6. Enable the LDAP server for authentication for specified tunnel-group.

pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: (CRYPT)*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.1.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com" -w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"

Verify

Test VPN Access

Anyconnect is configured to connect to 192.168.1.250. Log in is username cisco and password pass1.
After authentication the correct banner is used.

The correct split ACL is sent (ACL1 defined on ASA).

The Anyconnect interface is configured with IP: 10.1.1.1 and netmask 255.255.255.128. The domain is domain1.com and DNS server is 10.6.6.6.
On the ASA, user cisco has received IP: 10.1.1.1 and is assigned to group policy POLICY1.

ASA# show vpn-sessiondb detail anyconnect

Session Type: AnyConnect Detailed

Username : cisco Index : 29
**Assigned IP : 10.1.1.1** Public IP : 192.168.1.88
Protocol : AnyConnect-Parent SSL-Tunnel
License : AnyConnect Premium
Encryption : RC4 Hashing : none SHA1
Bytes Tx : 10212 Bytes Rx : 856
Pkts Tx : 8 Pkts Rx : 2
Pkts Tx Drop : 0 Pkts Rx Drop : 0

**Group Policy : POLICY1** Tunnel Group : RA
Login Time : 10:18:25 UTC Thu Apr 4 2013
Duration : 0h:00m:17s
Inactivity : 0h:00m:00s
NAC Result : Unknown
VLAN Mapping : N/A VLAN : none

AnyConnect-Parent Tunnels: 1
SSL-Tunnel Tunnels: 1

AnyConnect-Parent:
Tunnel ID : 29.1
Public IP : 192.168.1.88
Encryption : none TCP Src Port : 49262
TCP Dst Port : 443 Auth Mode : userPassword
Idle Time Out: 30 Minutes Idle TO Left : 29 Minutes
Client Type : AnyConnect
Client Ver : 3.1.01065
Bytes Tx : 5106 Bytes Rx : 788
Pkts Tx : 4 Pkts Rx : 1
Pkts Tx Drop : 0 Pkts Rx Drop : 0

SSL-Tunnel:
Tunnel ID : 29.2
Assigned IP : 10.1.1.1 Public IP : 192.168.1.88
Encryption : RC4 Hashing : SHA1
Encapsulation: TLSv1.0 TCP Src Port : 49265
TCP Dst Port : 443 Auth Mode : userPassword
Idle Time Out: 30 Minutes Idle TO Left : 29 Minutes
Client Type : SSL VPN Client
Client Ver : Cisco AnyConnect VPN Agent for Windows 3.1.01065
Bytes Tx : 5106 Bytes Rx : 68
Pkts Tx : 4 Pkts Rx : 1
Pkts Tx Drop : 0 Pkts Rx Drop : 0

Filter Name : AAA-user-cisco-E0CF3C05

NAC:
Reval Int (T): 0 Seconds Reval Left(T): 0 Seconds
SQ Int (T) : 0 Seconds EoU Age(T) : 17 Seconds
Hold Left (T): 0 Seconds Posture Token:

Also, the dynamic access-list is installed for that user:

ASA# show access-list AAA-user-cisco-E0CF3C05
access-list AAA-user-cisco-E0CF3C05; 1 elements; name hash: 0xf9b6b75c (dynamic)
access-list AAA-user-cisco-E0CF3C05 line 1 extended permit
ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
(hitcnt=0) 0xf8010475

Debugs
After you enable debugs, you can track each step of the WebVPN session.

This example shows LDAP authentication along with attribute retrieval:

ASA# show debug
debug ldap enabled at level 255
debug webvpn anyconnect enabled at level 254
ASA#

[63] Session Start
[63] New request Session, context 0xbbe10120, reqType = Authentication
[63] Fiber started
[63] Creating LDAP context with uri=ldap://192.168.11.10:389
[63] Connect to LDAP server: ldap://192.168.11.10:389, status = Successful
[63] supportedLDAPVersion: value = 3
[63] Binding as Manager
[63] Performing Simple authentication for Manager to 192.168.11.10
[63] LDAP Search:
  Base DN = [DC=test-cisco,DC=com]
  Filter = [uid=cisco]
  Scope = [SUBTREE]
[63] User DN = [uid=cisco,ou=People,dc=test-cisco,dc=com]
[63] Server type for 192.168.11.10 unknown - no password policy
[63] Binding as cisco
[63] Performing Simple authentication for cisco to 192.168.11.10
[63] Processing LDAP response for user cisco
[63] Authentication successful for cisco to 192.168.11.10
[63] Retrieved User Attributes:
  cn: value = John Smith
  givenName: value = John
  sn: value = cisco
  uid: value = cisco
  uidNumber: value = 10000
  gidNumber: value = 10000
  homeDirectory: value = /home/cisco
  mail: value = jsmith@dev.local
  objectClass: value = top
  objectClass: value = posixAccount
  objectClass: value = shadowAccount
  objectClass: value = inetOrgPerson
  objectClass: value = organizationalPerson
  objectClass: value = person
  objectClass: value = CiscoPerson
  loginShell: value = /bin/bash

Important! Custom LDAP attributes are mapped to ASA attributes as defined in ldap attribute-map:

[63] CiscoBanner: value = This is banner 1
  mapped to Banner1: value = This is banner 1
[63] CiscoIPAddress: value = 10.1.1.1
  mapped to IETF-RADIUS-Framed-IP-Address: value = 10.1.1.1
[63] CiscoIPNetmask: value = 255.255.255.128
  mapped to IETF-RADIUS-Framed-IP-Netmask: value = 255.255.255.128
[63] CiscoDomain: value = domain1.com
  mapped to IPSec-Default-Domain: value = domain1.com
[63] CiscoDNS: value = 10.6.6.6
  mapped to Primary-DNS: value = 10.6.6.6
[63] CiscoACLIn: value = ip:inacl#1=permit
  ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
  mapped to Cisco-AV-Pair: value = ip:inacl#1=permit
  ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
[63] CiscoSplitACL: value = ACL1
  mapped to IPSec-Split-Tunnel-List: value = ACL1
CiscoSplitTunnelPolicy: value = 1
mapped to IPSec-Split-Tunneling-Policy: value = 1
CiscoGroupPolicy: value = POLICY1
mapped to IETF-Radius-Class: value = POLICY1
mapped to LDAP-Class: value = POLICY1
userPassword: value = (SSHA)5s81Fmi/9aG/WfPSy3lGmw1ORI4lywWC
ATTR_CISCO_AV_PAIR attribute contains 68 bytes
Fiber exit Tx=315 bytes Rx=907 bytes, status=1
Session End

The LDAP session is finished. Now, ASA processes and applies those attributes.

The dynamic ACL is created (based on ACE the entry in Cisco-AV-Pair):
webvpn_svc_parse_acl: processing ACL: name: 'AAA-user-cisco-E0CF3C05', list: YES, id -1
webvpn_svc_parse_acl: before add: acl_id: -1, acl_name: AAA-user-cisco-E0CF3C05
webvpn_svc_parse_acl: after add: acl_id: 5, acl_name: AAA-user-cisco-E0CF3C05, refcnt: 1

The WebVPN session proceeds:
webvpn_rx_data_tunnel_connect
CSTP state = HEADER_PROCESSING
http_parse_cstp_method()
...input: 'CONNECT /CSCOSSLC/tunnel HTTP/1.1'
webvpn_cstp_parse_request_field()
...input: 'Host: 192.168.1.250'
Processing CSTP header line: 'Host: 192.168.1.250'
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
...input: '
webvpn_cstp_parse_request_field()
webvpn_cstp_parse_request_field()  
...input: 'X-CSTP-Full-IPv6-Capability: true'
webvpn_cstp_parse_request_field()  
...input: 'X-DTLS-Master-Secret: F5ADDD0151261404504FC3B165C3B68A90E51A1C8EB7EA9B2FE70F1EBB8E10929FFD79650B07E218EC8774678CDE1FB5E'
webvpn_cstp_parse_request_field()  
...input: 'X-DTLS-Master-Secret: F5ADDD0151261404504FC3B165C3B68A90E51A1C8EB7EA9B2FE70F1EBB8E10929FFD79650B07E218EC8774678CDE1FB5E'
webvpn_cstp_parse_request_field()  
...input: 'X-DTLS-CipherSuite: AES256-SHA:AES128-SHA:DES-CBC3-SHA:DES-CBC-SHA'
webvpn_cstp_parse_request_field()  
...input: 'X-DTLS-Accept-Encoding: lzs'
webvpn_cstp_parse_request_field()  
...input: 'X-DTLS-Header-Pad-Length: 0'
webvpn_cstp_parse_request_field()  
...input: 'X-CSTP-Accept-Encoding: lzs, deflate'
webvpn_cstp_parse_request_field()  
...input: 'X-CSTP-Protocol: Copyright (c) 2004 Cisco Systems, Inc.'

Next, address assignment occurs. Notice there is no IP pool defined on the ASA. If LDAP does not return the CiscoIPAddress attribute (which is mapped to IETF-Radius-Framed-IP-Address and used for IP address assignment), the configuration would fail at this stage.

Validating address: 10.1.1.1
CSTP state = WAIT_FOR_ADDRESS
webvpn_cstp_accept_address: 10.1.1.1/255.255.255.128
webvpn_cstp_accept_ipv6_address: No IPv6 Address
CSTP state = HAVE_ADDRESS

The WebVPN session completes:

SVCC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup

SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED

ASA Separate Authentication and Authorization

Sometimes it is better to separate authentication and authorization process. For example, use password authentication for locally defined users; then, after successful local authentication, retrieve all user attributes from LDAP server:

SVCC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
ASA Attributes from LDAP and Local Group

It's important to understand the difference between LDAP attributes and RADIUS attributes.

When you use LDAP, ASA does not allow mapping to any radius attribute. For example, when you use RADIUS, it is possible to return the cisco-av-pair attribute 217 (Address-Pools). That attribute defines a locally configured pool of IP addresses that are used to assign IP addresses.

With LDAP mapping, it is impossible to use that specific cisco-av-pair attribute. The cisco-av-pair attribute with LDAP mapping can be used only to specify different types of ACLs.

These limitations in LDAP prevent it from being as flexible as Radius. To workaround this locally defined group policy can be created on the ASA with attributes which can not be mapped from ldap (like Address-Pools). Once the LDAP user is authenticated, they are assigned to that group policy (in our example POLICY1) and the non user-specific attributes are retrieved from the group-policy.

The full attribute list supported by LDAP mapping can be found in this document: Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6

You can compare to the full list of RADIUS VPN3000 attributes supported by ASA; refer to this document: Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6

Refer to this document for a full list of RADIUS IETF attributes supported by ASA: Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6

ASA and LDAP with Certificate Authentication
ASA does not support LDAP certificate attribute retrieval and binary comparison with certificate provided by Anyconnect. That functionality is reserved for Cisco ACS or ISE (and only for 802.1x supplicants) because VPN authentication is terminated on a network access device (NAD).

There is another solution. When user authentication uses certificates, ASA performs certificate validation and can retrieve LDAP attributes based on specific fields from certificate (for example, CN):

```
SVC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup
SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED
```

After the user certificate is validated by ASA, LDAP authorization is performed and user attributes (from CN field) are retrieved and applied.

**Debugs**

User certificate has been used: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL

Certificate mapping is configured to map that certificate to the RA tunnel-group:

```
SVC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup
SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED
```

Certificate validation and mapping:

```
ASA# show debug
debug ldap enabled at level 255
debug webvpn anyconnect enabled at level 254
debug crypto ca enabled at level 3
debug crypto ca messages enabled at level 3
debug crypto ca transactions enabled at level 3
Apr 09 2013 17:31:32: %ASA-7-717025: Validating
```
certificate chain containing 1 certificate(s). Apr 09 2013 17:31:32: %ASA-7-717029: Identified client certificate within certificate chain. serial number: 00FE9C3D61E131CDB1, subject name: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL. Apr 09 2013 17:31:32: %ASA-6-717022: Certificate was successfully validated. Certificate is resident and trusted, serial number: 00FE9C3D61E131CDB1, subject name: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL. Apr 09 2013 17:31:32: %ASA-6-717028: Certificate chain was successfully validated with revocation status check. Apr 09 2013 17:31:32: %ASA-6-717028: Certificate chain was successfully validated with revocation status check. Apr 09 2013 17:31:32: %ASA-7-717036: Looking for a tunnel group match based on certificate maps for peer certificate with serial number: 00FE9C3D61E131CDB1, subject name: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL, issuer_name: cn=TAC,ou=RAC,o=TAC,l=Warsaw,st=Maz,c=PL. Apr 09 2013 17:31:32: %ASA-7-717038: Tunnel group match found. Tunnel Group: RA. Peer certificate: serial number: 00FE9C3D61E131CDB1, subject name: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL, issuer_name: cn=TAC,ou=RAC,o=TAC,l=Warsaw,st=Maz,c=PL.

Extraction of username from certificate and authorization using LDAP:

Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN client certificate has been requested. [Request 53]Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN client certificate has been requested. [Request 53]Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN client certificate has been requested. [Request 53]Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN client certificate has been requested. [Request 53]Apr 09 2013 17:31:32: %ASA-7-113004: AAA user authorization Successful : server = 192.168.11.10 : user = test1Apr 09 2013 17:31:32: %ASA-6-113004: AAA user authorization Successful : server = 192.168.11.10 : user = test1Apr 09 2013 17:31:32: %ASA-7-113004: AAA user authorization Successful : server = 192.168.11.10 : user = test1Apr 09 2013 17:31:32: %ASA-7-113004: AAA user authorization Successful : server = 192.168.11.10 : user = test1

Attributes retrieval from LDAP:


John Smith

John Smith
Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoSplitACL = ACL1**

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoSplitTunnelPolicy = 1**

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoGroupPolicy = POLICY1**

Cisco mapped attributes:

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.cisco.grouppolicy = POLICY1

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.cisco.ipaddress = 10.1.1.1

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.cisco.username = test1

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.cisco.username1 = test1

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.cisco.username2 = test1

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.cisco.tunnelgroup = RA

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88, Connection AnyConnect: The following DAP records were selected for this connection: DfltAccessPolicy

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88, Connection AnyConnect: The following DAP records were selected for this connection: Group <POLICY1> User <test1> IP <192.168.1.88> AnyConnect parent session started.

**Secondary Authentication**

If two-factor authentication is required, it is possible to use token password along with LDAP authentication and authorization:

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88, Connection AnyConnect: The following DAP records were selected for this connection: DfltAccessPolicy

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88, Connection AnyConnect: The following DAP records were selected for this connection: Group <POLICY1> User <test1> IP <192.168.1.88> AnyConnect parent session started.

Apr 09 2013 17:31:32: %ASA-6-113039: Group <POLICY1> User <test1> IP <192.168.1.88> AnyConnect parent session started.

Then, the user must provide a username and password from RSA (something the user has—a token), along with LDAP username/password (something the user knows). It is also possible to use a username from the certificate for secondary authentication. For more information about double authentication, refer to the Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6.

**Related Information**

- [Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6](#)
- [OpenLDAP Software 2.4 Administrator’s Guide](#)
- [Private Enterprise Numbers](#)
- [Technical Support & Documentation - Cisco Systems](#)