Implementing Directory Integration and Single Sign-On with Cisco UC

BRKUCC-3420

Paulo Jorge Correia
July 12, 2011
Abstract

In this session we will cover different corporate User directory solutions (based on LDAP protocol) like AD, ADAM, AD LDS, OpenLDAP and we will explore the different ways that they will integrate into Cisco collaboration solution.

We will explore the most common user directory deployment models (single domain, multiple domains, single directory or multiple directory structure) for each solution.

We will also explore SSO (Single Sign-On) subject and it’s integration with Cisco Collaboration portfolio using an Open Web Single Sign-On architecture based on OpenAM.

In the end of this presentation, the attendees will understand how these User Directories solutions and SSO Solution interact with the Cisco Collaboration portfolio, and will understand the deployment models that should be used in each situation.

Attendees should have a good working knowledge of user directory solutions based on LDAP, good understanding of SSO concepts and know Cisco Collaboration Portfolio.
Today’s Challenges for Directories and Authentication & Authorization Services

Understand LDAP

Understand What Cisco Collaboration Portfolio Expects from Users Directories Services

User Directories Frontends to AD

Microsoft ADAM / AD LDS
  OpenLDAP

Single Sign-On a New Approach to Authentication & Authorization Services

Key Takeaways and Q&A
Today’s Challenges for Directories and Authentication & Authorization Services
What Is a User Directory?

- A directory service is a publicly available database of structured information.

- The most common example of a directory service is your local Yellow Pages - it contains names, addresses, etc.

- All indexed in a manner that is easily browseable or searchable.

- The service can be categorized as 'write-once-read-many-times'.
Where to Store the User Information?

- Organizations need a Single place to Store information.
- For Administrators and Support, Organization to add and update information about users, only once in a single place.
- For users to look other users details, and to provide user information to different applications.
What Do We See Today in Organizations?

- Almost all the organizations already have an Active directory infrastructure.

- Organizations consolidates user passwords in AD, but they don’t fully use it as a user directory source for other applications.

In the Rest of the Presentation We Are Going to Cover Different Architecture to Simplify the Complexity of AD Deployments

Check Appendix A for the Basic Concepts of Active Directory
But Sometimes It Gets Very Complicated...

Complex AD Deployments that Aren’t Prepare to Consolidate User Information.

Forest 1
GC/DC Domain webex.com

Forest 2
GC/DC Domain Cisco.com

Forest 3
GC/DC Domain Tandberg.com

Emerg.cisco.com
Applications Are Starting to Be More Dependent on LDAP Systems

We Start to Have More and More Applications, that Need to Consult the LDAP Database in Every Task that They Execute… That Is Too Heavy for the AD DC, They Aren’t Designed for That Kind of Traffic
Users Ask to Provide Authentication Only Once

With So Many Systems and Applications that Need User Directory Information and Authentication, Users Are Complain More and More Every Day…

*Why Should I Have to Provide Credentials in Every System that I Use?*

*Why Do I Need to Update All the Applications When I Change My Password?*
**Pop Quiz!!!!!**

**Question:** What is the best topology to have Users directories, authentication and authorization systems from an users and system admins perspective?

A. To have usernames and password in each application for security reasons

B. To have a consolidated directory services and every time that users connect to an application, their credentials are requested

C. Organizations don’t care about applications credentials, applications shouldn’t ever ask for it, they have a very secure network 😊

D. Have a consolidated user directory, where users provide credentials only once during the working session to the first application used.
Understand LDAP
What Is LDAP?

- It was created by IETF in the early 90’s

- The Lightweight Directory Access Protocol (LDAP) is an open industry standard that has evolved to meet User Directories needs.

- LDAP defines a standard methods for accessing and updating information in a directory.

- LDAP has gained wide acceptance as the directory access method in the Internet and is therefore also becoming strategic within corporate intranets.

- LDAP is characterized as a 'write-once-read-many-times' service.
Different Types of LDAP Implementations

eDirectory: This is Novell's implementation of directory services.

Active Directory: Microsoft's modern directory service for Windows, originating from the X.500 directory it created for use in Exchange Server.

Sun Java System Directory Server: Sun Microsystems directory service offering

OpenLDAP: Derived from the original University of Michigan reference LDAP implementation but significantly evolved.
Different Representations of a Directory Tree

LDAP Directory Tree (Internet Naming)

LDAP Directory Tree (Traditional Naming)

Paulo Correia
Example of a LDAP Directory Information Tree

**DN** – Distinguished Names  
**RDN** – Relative Distinguished Name  
**OU** – Organization Unit  
**CN** – Common Name  
**DC** – Domain Component

**OU=devices**

**OU=people**

**CN=Paulo Correia**

**DN:** `CN=Paulo Correia, OU=people, DC=cisco, DC=com`

- **CN:** Paulo Correia  
- **objectClass:** person  
- **SN:** Correia  
- **telephoneNumber:** 111-1234
LDAP Operations

- **Start TLS** — Use the LDAPv3 TLS extension for a secure connection
- **Bind** — Authenticate and specify LDAP protocol version
- **Search** — Search for and/or retrieve directory entries
- **Compare** — Test if a named entry contains a given attribute value
- **Add** a new entry
- **Delete** an entry
- **Modify** an entry
- **Modify Distinguished Name (DN)** — Move or rename an entry
- **Abandon** — Abort a previous request
- **Extended Operation** — Generic operation used to define other operations
- **Unbind** — Close the connection (not the inverse of Bind)
ObjectClass

- objectClasses group sets of attributes.
- objectClasses are defined inside schemas.
- Each record represents an object, and the attributes associated with that object are defined according to its objectClass.

Examples:
- Organization (needs a name and address)
- Person (needs name, email, phone & address)
- Course (needs a CRN, instructor, mascot)
- Cookie (needs name, cost & taste index)
LDAP Operations

Schema

The Schema Defines the *Attribute Types* that Directory Entries Can Contain. The Contents of the Entries in a Subtree Is Governed by a Schema. The Schema Defines Object Classes and Entry Must Have an ObjectClass Attribute (e.g. a Person, Organization or Domain.)

**Examples:**

```
attributetype ( 0.9.2342.19200300.100.1.1
  NAME ( 'uid' 'userid' )
  DESC 'RFC1274: user identifier'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

```
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top
  DESC 'Abstraction of an account with POSIX attributes'
  MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
  MAY ( userPassword $ loginShell $ gecos $ description ) )
```
Typical LDIF Record for a User

LDIF Entries Are LDAP Data Interchange Format and Typically We Find the In Files to Be Imported/Exported to an LDAP System

A typical entry serialized in LDIF:

```ldif
dn: cn=Paulo Jorge Correia,dc=cisco,dc=com

cn: Paulo Jorge Correia

givenName: Paulo

sn: Correia

televisionNumber: +1 111 6689

televisionNumber: +1 333 1234

mail: paucorre@cisco.com

manager: cn=Zorela Sora,dc=cisco,dc=com

objectClass: inetOrgPerson

objectClass: organizationalPerson

objectClass: person

objectClass: top
```
Pop Quiz!!!!!

**Question:** In LDAP DIT (Directory Information Tree) what does OU stands for?

- **A** Operation Unit
- **B** Organization Users
- **C** Organization Unit
- **D** Organic User
Understand What Cisco Collaboration Portfolio Expects from Users Directories Services
CUCM Directory Architecture Without LDAP Integration

[Diagram showing the architecture with components like Cisco Unified CM Publisher, Corporate LDAP Directory, System Accounts, DB, DirSync, End User Accounts, IMS, Tomcat, AXL/SOAP, HTTP connections, and arrows indicating Administrator, End User, and Directory Lookup.]
CUCM Directory Architecture With Synchronization and Authentication Enabled

Cisco Unified CM Publisher

- System Accounts
- End User Accounts
- DirSync

Corporate LDAP Directory

Administrator

End User

Directory Lookup
CUCM Directory Architecture With Synchronization and Authentication for Big Deployment

Cisco Unified CM Publisher

- System Accounts
- End User Accounts
- DB
- DirSync
- IMS
- Tomcat
- AXL/SOAP

LDAP Frontend

- LDAP search
- LDAP Bind
- LDAP User Synchronization
- HTTP

Administrator

End User

Directory Lookup

CSF Client

HTTP

HTTP

HTTP

HTTP

HTTP Server
CUCM AD Synchronization

- **LDAP Synchronization** - Uses an internal tool called Cisco Directory Synchronization (DirSync) to synchronize a subset of CUCM user data (first name, last name, alias, phone number, and so on, either manually or periodically).

- **LDAP Authentication** - Some companies want the convenience of a single password for their applications.
CUCM AD Synchronization

- Synchronization agreements are limited to a single domain and not cascade to child domains. Separate Agreements are required (at least 1 per domain) up to 5 synchronization agreements.

- Each Agreement may specify up to 3 Directory servers for redundancy.

- Connections can use LDAP or be secured via LDAPS (SSL).

- Configurable synchronization schedule between 6 hours to 30 days. Also supports manual sync.
CUCM AD Synchronization

- Last name is a mandatory attribute. Users without a last name attribute configured are not synchronized.
- Users in the corporate directory with same user ID as existing Application Users are not imported.
- End Users must be unique based on the chosen directory attribute that is mapped to the Cisco Unified CM User ID field.
- End users cannot be manually added via web Administration pages when synchronization is enabled.
CUCM AD Authentication

- You can only authenticate against one domain
- In a Multiple domain structure you should point to the global catalog
- The PIN field continues to be managed in the local database.

### LDAP Authentication

**Status**

- Status: Ready

**LDAP Authentication for End Users**

- Use LDAP Authentication for End Users
- LDAP Manager Distinguished Name: `cn=Administrator,cn=Users,dc=emuclab,dc=org`
- LDAP Password:
- Confirm Password:
- LDAP User Search Base: `cn=Users,dc=emuclab,dc=org`

**LDAP Server Information**

<table>
<thead>
<tr>
<th>Host Name or IP Address for Server</th>
<th>LDAP Port</th>
<th>Use SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.33.130</td>
<td>3268</td>
<td></td>
</tr>
</tbody>
</table>

Add Another Redundant LDAP Server
Performance

- Initial synchronization for 60,000 User Accounts in Unified CM 7.1(x) is roughly 30 minutes.

- Initial synchronization for 60,000 User Accounts in Unified CM 8.0(1) is roughly 22 minutes.

- Subsequent synchronization operations may take more/less time depending on the number of changes between synchronization intervals.

- A maximum of 60,000 User Accounts per Unified CM Publisher are supported.
Unity Connection User Directories Integration

- We have three different approaches to user directories in Unity Connection
  - Importing users from a CSV file to the Connections database
  - Getting the user from CUCM database using AXL/SOAP
  - Connection to an LDAP engine using the same mechanisms and limitations discussed before in CUCM
Presence Server User Directories Integration

- CUP gets the user and password information from CUCM through the sync agent.
- CUP also provide AD directory integration for CUPC.
MeetingPlace User Directories Integration

- It integrates with Unified CM via Cisco AVVID XML Layer (AXL) Simple Object Access Protocol (SOAP) over secure HTTP (HTTPS).
- We will use SSO to populate users in the WebEx Cloud (it will be discussed in detail in the SSO section).
**Pop Quiz!!!!!**

**Question:** In CUCM how many domains can we synchronize with?

- **A** 5
- **B** 3
- **C** 1
- **D** 11
User Directories
Frontends to AD
Microsoft ADAM / AD LDS
What Is AD LDS/ADAM?


AD LDS Can Be Used in Conjunction with AD DS So that You Can Have a Central Location for Security Accounts (AD DS) and Another Location to Support the Application Configuration and Directory Data (AD LDS).
What Is AD LDS/ADAM?

How Do We Handle the Single Password with This Implementation?

- A special user proxy object in AD-LDS maps to a regular Active Directory user account.
- The user proxy does not have an actual password stored in the AD-LDS object itself.
- When the application performs its normal bind operation, it checks the ID locally, but checks the password against Active Directory under the covers.
How Can We Use ADAM/AD LDS to Create a Frontend to Multiple AD Domains, Trees, Forests?

As We Saw Before in the Cisco Collaboration Portfolio We Can’t Support Any Deployment Outside a Single Forest.

With Solutions Like AD LDS/ADAM We Can Create an LDAP Frontend to the Different AD Domain Services.
How Do We Implement This?

1. Create outgoing trust from the domain that hosts the ADAM instance to the domains in the different forests.

2. Copy the Schema from the different forests to the AD LDS/ADAM instance.

3. Extend the AD-LDS Schema with the USER-PROXY object.

4. Import the users from AD DC to AD LDS.

5. Creating an user in AD LDS for allowing synchronization and authentication for the Cisco Collaboration Applications.

6. Configure Cisco Collaboration application to get the user information from AD LDS/ADAM.

Note: For This Solution to Work You Will Need to Change the Search Filter in CUCM from Objectclass User to UserProxy Created in Step 3.

Check Appendix B for More Detailed Information.

**Question:** With Bind redirect feature in ADAM/AD LDS where are the users passwords stored?

- **A** ADAM / AD LDS
- **B** Active Directory Domain Services
- **C** Text file
- **D** External database like MS SQL, Oracle, etc.
User Directories
Frontends to AD
OpenLDAP
What Is OpenLDAP?

- OpenLDAP 2 is compliant with the core LDAPv3 specifications.
- OpenLDAP is available for multiple platforms, including Linux, Solaris, Mac OS and Windows
- The OpenLDAP project is a continuation of the original University of Michigan LDAP server.
OpenLDAP Working Model

OpenLDAP Implementation

- Slapd – standalone server daemon
- Slurpd – replication daemon (deprecated from version 2.3 forward)
- Libraries including Java libraries

Diagram:
- Client → Frontend (Request) → Backend (Response)
  Frontend: slapd
  Backend: bdb, dbm, hdb
OpenLDAP Topology for a Very Big Deployment
How Can We Use OpenLDAP to Create a Frontend to Multiple AD Domains, Trees, Forests?

Assuming that We Have the Same AD Structure as We Had with ADAM/AD LDS

Forest 1
- GC/DC
- Domain: webex.com

Forest 2
- GC/DC
- Domain: Cisco.com

Forest 3
- GC/DC
- Domain: Tandberg.com

OpenLDAP SASL

Cisco UC Components
How Do We Implement This?
What Are the Components Required?

- We need to install and configure OpenLDAP to host the users information that is going to be imported from AD DC’s.
- We need to get the users information from the different AD domains (single or multiforest) to OpenLDAP, and that can be achieved using LSC (LDAP Synchronizer Project).
- When we import the users their password should have the format {SASL}username@domain.
- Configure SASL to use Kerberos and provide information of the different KDC (Kerberos Domain Controllers).

Check Appendix C for More Detailed Information
Why Do We Need LSC?

- The conversion of user information from AD to OpenLDAP isn’t a trivial task, there are too many differences in the LDAP structure of both products.

- So you should use LSC [http://lsc-project.org](http://lsc-project.org) or similar product for synchronization

- You should configure LSC need to run periodically to keep the OpenLDAP information updated.

- LSC should run in sync with the CUCM and other Cisco collaboration products (for example if CUCM synchronize with OpenLDAP runs at 2.00 AM, LSC should run before than that.)
Why Do We Need SASL?

- This method of authentication relay requests between OpenLDAP and AD.

- SASL (Simple Authentication and Security Layer, defined in RFC 2222 that become obsolete with RFC 4422) have multiple ways of doing authentication, but since we have multiple domains we recommend to use Kerberos.

- OpenLDAP to send authentication requests of an user to AD, i.e. the bind operation of LDAP through saslauthd.
How to Access Through a GUI to OpenLDAP?

From the Linux GUI Interface You Can Install and Use Apache Directory Studio.

http://directory.apache.org/studio/

It Provides You with All the LDAP Operations that You Have from the Command Line.
Pop Quiz!!!!!

**Question:** In which University was OpenLDAP developed?

A. Minho
B. Michigan
C. MIT
D. Cambridge
Single Sign-On a New Approach to Authentication & Authorization Services
SSO Objective

User Needs to Authenticate to the Desktop, Mobile or Server Once and All Other Applications Within an Endpoint Should Use That Authentication to Provide Access.

The End Goal Is to Provide a Seamless User Experience Across All UC Services and I Get Access to All Systems Without Being Repeatedly Prompted to Log In Again at Each of Them.
Proposed Authorization & Authentication Services Solution

- Integration of the *Cisco Identity Management System* (IMS) with the *Open Web Single Sign-On architecture*, which is an open source initiative by Sun Microsystems (https://opensso.dev.java.net/)

- Oracle has announced that OpenSSO would no longer be their strategic product, after its acquisition of Sun Microsystems. OpenSSO continues to be developed and supported by ForgeRock under the name of *OpenAM*
OpenSSO

- The OpenSSO code is released under the Common Development and Distribution License (CDDL) [http://www.sun.com/cddl/](http://www.sun.com/cddl/).

- OpenAM supports several open federation technologies including the Security Access Markup Language (SAML) versions 1 and 2, WS-Federation, and the Liberty Alliance Project Identity Federation Framework (Liberty ID-FF).

- **SAML** Standard for passing credentials between different Internet domains that have their own authentication systems.

  [http://www.oasis-open.org/committees/security](http://www.oasis-open.org/committees/security)
Steps to Deploy

- Deploy & configure a server to run ForgeRock OpenAM
- Provision OpenAM access service in Active directory
- Execute new CLI Commands on CUCM to enable SSO

Check Appendix D for More Detailed Information
CUCM SSO Flow

User Authenticates to Enterprise AD Using Smartcard/Bio-Metric or Password

401 WWW-Authenticate: AM Server

SSO Cookie

Redirect with SSO Cookie

https://<cucm-ip>/ucm-webapp

redirect to the application

Validate SSO Cookie

Using TGT Request Service Ticket for OpenAM Server

TGT: Ticket Granting Ticket
ST: Service Ticket

TGT:

ST:

Validate SSO Cookie

Store in a Secure Container

Policy Agent

Web App (User Options)

Tomcat

CUCM

Active Directory

(Browser Configured with Integrated Windows Login)
CUCIMOC SSO Flow

User Authenticates to Enterprise AD Using Smartcard/Bio-Metric or Password

401 WWW-Authenticate: Negotiate
Authorize: Negotiate with ST

302 Redirect to OpenAM Server
Using TGT Request Service Ticket for OpenAM Server

Validate SSO Cookie

Redirect with ST to https://<cucimoc>/App/3uds/GetDeviceList()

Access to Application Returns Device List

Store in a secure container

TGT: Ticket Granting Ticket
ST: Service Ticket

UC Manager 8.5

CTI Manager

CMCIP / UDS

Policy Agent

Tomcat

OpenAM Client SDK

IMS Extensions

Active Directory

TGT
User Authenticates to Enterprise AD Using Smartcard/Bio-Metric or Password

Using TGT Request Service Ticket for OpenAM Server

Decrypt ST and Request Username

Service Request Using ST

Pass ST

TGT: Ticket Granting Ticket
ST: Service Ticket

CUCIMOC CSF 8.5
Windows Desktop

UC Manager 8.5

CTI Manager

Tomcat
Policy Agent
CCMCIP / UDS

IMS Extensions
OpenAM Client SDK

TGT
Active Directory

ST
UC SSO Flow

Active Directory

Validate Credentials

Validate Cookie and Cache

Web App

Policy Agent

Allow Access

Web App

Policy Agent

Allow Access

CUP

Validate Cookie and Cache

Browser

Login Request

SSO Credentials

CUP User WebApps with SSO Cookie

CUCM

Store in Secure Container

Validates in Secure Container

Store in Secure Container

SSO Achieved

Authenticaton Across Products

SSO Achieved

Across UC Products
WebEx SSO
What Do We Need to Know About SSO Federation?

- Users do not need to remember WebEx usernames or password
- **No WebEx passwords are stored or transmitted**
- Utilizes WebEx Federated Authentication Service (FAS)
- Requires an Identity and Access Management (IAM) system that conforms to:
  - Security Assertion Markup Language (SAML) 1.1 or 2.0
  - WS-Federation 1.0
- Webex is referred to a Service Provider (SP)
- IAM Certificate needs to be uploaded into WebEx
WebEx SSO

What SSO Engines Are Supported by FAS?

The IAM System Should Function as a SAML or WS-Federation for the Identity Provider (IdP). The IAM System Should Be Able to Produce SAML Assertions or WS-Fed Tokens Digitally Signed with X.509 Certificates. The WebEx FAS Has Been Tested with the Following Commercial IAM Systems:

- CA SiteMinder
- Ping Identity PingFederate
- **Sun Microsystems OpenSSO Enterprise**
- Microsoft Windows Server ADFS and Geneva
- Novell Identity Manager
- IBM Tivoli Federated Identity Manager
- Siemens IT Solutions DirX
- TriCipher Armored Credential System
- Fugen Solutions
- Cloud-Identity
WebEx SSO
IdP Initiated SSO

Active Directory -> 0. Windows Login

0. Windows Login

1. Challenge for Credential

2. User Login

3. Validate Credentials

4. Select WebEx Service

5. Signed Response

6. POST Signed Response

7. Verify User

8. Establish WebEx Host Session & Redirect

User DB -> FAS (Federated Authentication Service)

WebEx Target Page -> User DB

WebEx Cloud Services

IDMS

User Web Browser
WebEx SSO
SP Initiated SSO

1. User Clicks “Host Log In”
2. Redirect with AuthnRequest
3. Challenge for Credential
4. User Login
5. Validate Credentials
6. Signed Response
7. POST Signed Response
8. Verify User
9. Establish WebEx Host Session & Redirect

User DB
FAS (Federated Authentication Service)
WebEx Cloud Services
WebEx Meeting Page

User Web Browser
IDMS
Active Directory

0. Windows Login

© 2011 Cisco and/or its affiliates. All rights reserved. Cisco Public
WebEx SSO
WebEx FAS Capability Matrix

The Following Table Summarizes WebEx FAS Functionality When Used with Each of the Supported Federated SSO protocols.

<table>
<thead>
<tr>
<th>Feature</th>
<th>WS-Federation 1.0</th>
<th>SAML 1.1</th>
<th>SAML 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdP Initiated SSO</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SP Initiated SSO</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Productivity Tools SSO</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Integrated Windows Authentication</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic Account Provisioning</td>
<td>Yes</td>
<td>Requires custom Java programming</td>
<td>Yes</td>
</tr>
<tr>
<td>Single Logout</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
WebEx Connect
Single Sign On - Architecture

0. Windows Login

4. Validate Credentials

3. Credential Challenge

5. SAML Assertion *

2. Authentication Request

1. Federated SSO Request

6. Federated SSO Result/Page

CAS (Connect Authentication Service) Validates the User Specified in the SAML Assertion with the Connect Database

- If it's a first time the user connects, CAS provisions the new user in the Connect database.
- CAS returns a validated user token to the Connect Client
Question: Which protocol is required for on-premise federation with WebEx cloud service?

A) SASL
B) SSL
C) SAML
D) Kerberos
Key Takeaways and Q&A
Key Takeaways

- **A consolidated User Directory** is a requirement to successful deployment collaboration solutions.

- Before starting an implementation, you need to understand how user information is **stored** inside the Organization and how can you **access** to it.

- If you don’t have a consolidate User Directory, you should modify it to achieve that goal. If that isn’t an option, you can achieve that through a **LDAP frontend** to the AD servers existing in your organization.

- There are a lot of confusion in the market on what is Single Sign On. With SSO the different application should never ask for authentication after successful authentication in the first application.
Please Browse On-Site Cisco Store for Suitable Reading.
Complete Your Online Session Evaluation

- Receive 25 Cisco Preferred Access points for each session evaluation you complete.
- Give us your feedback and you could win fabulous prizes. Points are calculated on a daily basis. Winners will be notified by email after July 22nd.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.
- Don’t forget to activate your Cisco Live and Networkers Virtual account for access to all session materials, communities, and on-demand and live activities throughout the year. Activate your account at any internet station or visit www.ciscolivevirtual.com.
Visit the Cisco Store for Related Titles

http://theciscostores.com
Connect with Your Peers and Cisco

- Discuss business, IT, architecture, adoption and product topics with peers
  - conferencing, customer care, enterprise social software, IP communications, messaging, mobile applications, telepresence
- Interact with Cisco Product Managers, Technical Marketing Engineers and Services Consultants
- Learn about new product announcements
- Join the Collaboration User Group
  - Influence product direction
  - Access to early product releases and info
  - Get VIP perks and exclusive sessions at Cisco Live

Cisco Collaboration Community and User Group

Visit the Cisco Collaboration Booth (#1289) at Cisco Live:
Learn more about the community. Sign up for the user group. Pick up your polo shirt and badge ribbon (user group members only).
Thank you.
Appendix A
Active Directory

- Microsoft directory services
- Uses LDAP
- Uses Kerberos 5
- Uses DNS (not WINS)
- Scalable and available architecture
- Extends significantly but does not replace common X.500 schema elements

Mainly Used in Organizations for Centralized Password Control and Management
AD Terminologies and Topologies

- **Domain** is the core unit of logical structure in Active Directory. All objects which share a common directory database, trust relationship with other domain and security policies is known as Domain. Each domain stores information only about the objects that belong to that domain.
AD Terminologies and Topologies

- **Domain Controller** is a server on a Microsoft Windows network that is responsible for allowing host access to Windows domain resources.
**AD Terminologies and Topologies**

- **Trees** are collections of one or more domains that allow global resource sharing.
**Forest** is a collection of multiple trees that share a common global catalog, directory schema, logical structure, and directory configuration.
Global catalog is a distributed data repository that contains a searchable, partial representation of every object in every domain in a multidomain Active Directory forest.
Resource Forests in this topology the user accounts live in their native AD domain and it is created another Active Directory forest that stores applications specific information form all users in the organization.
How Do We Implement This?

Domain Trust Relationship

- For the authentication of the users to work, you need to have a trust between the domain where the ADAM instance is hosted, and the other domain(s) that host the user accounts. This trust can be a one-way trust if required (outgoing trust from the domain that hosts the ADAM instance to the domain(s) that host the user accounts). This way, the ADAM instance will be able to forward the authentication requests to DC’s in those account domains.
How Do We Implement This?
Copy the Schema from Each Domain to ADAM

- This process will need to be repeated for each domain that you need to synchronize to.
- With AD DS/LDS schema analyzer (ADSchemaAnalyzer.exe), load the schema from the domain that you want to import, then load the base schema from AD/LDS. After this step you should Mark all non-present elements as included, and create a ldif file with the result.

Finally Import the ldif file to AD-LDS with the LDIFDE Command
Refer to the Microsoft Documentation for Additional ldifde Options and Command Formats
http://support.microsoft.com/kb/237677

Example:
ldifde -i -s localhost:50000 -c CN=Configuration,DC=X #ConfigurationNamingContext -f diff-schema.ldf -j c:\windows\adam\logs
How Do We Implement This?
Extend the AD-LDS Schema with the USER-PROXY Objects

- The object for the proxy authentication needs to be created and the object class ‘user’ should not be used. The object class being created, userProxy, is what allows the bind redirection. The object class detail needs to be created in a ldif file, and we can use the default example provided by Microsoft (MS-UserProxy-Cisco.ldf) with some changes.

- Then you need to Import the new object class to AD-LDS

```
Idifde -i -s localhost:50000 -c CN=Configuration,DC=X #ConfigurationNamingContext -f MS-UserProxy-Cisco.ldf -j c:\windows\adam\logs
```

Example:

```
dn: CN=User-Proxy,CN=Schema,CN=Configuration,DC=X
changeType: ntdsSchemaAdd
objectClass: top
objectClass: classSchema
cn: User-Proxy
subClassName: top
governsID: 1.2.840.113556.1.5.246
schemaIDGUID: bxjWYlbzmEiwrWU1r8B2IA==
rDNAttID: cn
showInAdvancedViewOnly: TRUE
adminDisplayName: User-Proxy
adminDescription: Sample class for bind proxy implementation.
objectClassCategory: 1
LDAPDisplayName: userProxy
systemOnly: FALSE
possSuperiors: organizationalUnit
possSuperiors: container
possSuperiors: organization
defaultSecurityDescriptor:
    D:(OA;;CR;ab721a53-1e2f-11d0-9819-00aa0040529b;;PS)S:
defaultHidingValue: TRUE
defaultObjectCategory: CN=User-Proxy,CN=Schema,CN=Configuration,DC=X
systemAuxiliaryClass: msDS-BindProxy
systemMayContain: userPrincipalName
systemMayContain: givenName
systemMayContain: middleName
systemMayContain: sn
systemMayContain: manager
systemMayContain: department
```

----------
How Do We Implement This?
Import the Users from AD DC to AD LDS

- This step needs to be repeated for each domain that needs to synchronize. Starting with the original MS-AdamSyncConf.xml provided by Microsoft, create a XML file for each domain that needs to be synchronized and modify the file with the details specific to each domain something like what is shown below.

In This File, the Following Tags Should Be Replaced to Match the Domain

- `<source-ad-name>` - Use the host name of the domain
- `<source-ad-partition>` - Use the root partition from the source AD DC that you want to import from
- `<base-dn>` - Choose the container from which to import from.

```xml
<?xml version="1.0"?>
<doc>
  <configuration>
    <description>Adam-Sync1</description>
    <security-mode>object</security-mode>
    <source-ad-name>ad2k8-1</source-ad-name>
    <source-ad-partition>dc=cisco,dc=com</source-ad-partition>
    <source-ad-account></source-ad-account>
    <account-domain></account-domain>
    <target-dn>dc=cisco,dc=com</target-dn>
    <query>
      <base-dn>dc=cisco,dc=com</base-dn>
      <object-filter>(&amp;(objectCategory=Person)(objectClass=User))</object-filter>
      <attributes>
        <include>objectSID</include>
        <include>mail</include>
        <include>userPrincipalName</include>
        <include>middleName</include>
        <include>manager</include>
        <include>givenName</include>
        <include>sn</include>
        <include>department</include>
        <include>telephoneNumber</include>
      </attributes>
    </query>
  </configuration>
</doc>
```

Finally Get the Users:

ADAMSync /install localhost:50000
c:\windows\ADAM\AdamSyncConf1.xml /log
c:\windows\adam\logs\install.log

ADAMSync /sync localhost:50000 "dc=cisco,dc=com"
/log c:\windows\adam\logs\sync.log
How Do We Implement This?
Creating the User in AD LDS for Allowing Synchronization and Authentication

- Open ADSI Edit, from the Administrator tools in the startup menu
- Choose File, connection (or Action, Connect To) and connect to your AD LDS detail.
- Create a new object type user, and set the password.
- Browse it user properties and enable that user, setting msDS-UserAccountDisable to False.
- And finally add it to the administrator group.
How Do We Implement This?

Configuring Unified CM

- ADAM/AD LDS synchronization and authentication is supported in Unified CM version 7.1(3) and later.
- Choose in LDAP System type Microsoft Active Directory Application Mode
- Do not choose UserPrincipalName because Unified CM will not allow you to specify a user search base when you have UPN.
How Do We Implement This?

LDAP Filters in Unified CM

- The object class User is no longer being used; therefore, the ldap filter needs to be changed to use userProxy instead of User. Prior to Unified CM 8.0 there was no ability to modify the default filter:

\[
(\&(\text{objectclass} = \text{user})(!\text{(objectclass} = \text{Computer}))(!\text{(msDS-UserAccountDisabled} = \text{TRUE})))
\]

- To modify this filter on Unified CM release prior to 8.0, the administrator needs to use the AXL SOAP Toolkit to change the (Objectclass=user) to (Objectclass=userProxy). Refer to the document “User filtering for directory synchronization and authentication”. With Unified CM 8.0 and later under LDAP configuration there is an option for LDAP Custom Filter.

![LDAP Filter Configuration](image)
Appendix C
How Do We Implement This?

What Are the Components Required?

- We require OpenLDAP install on the Linux Box
- We need to create the top level domain for the user (ex: cisco.com)
- We need to get the users information from the different Ad domains to OpenLDAP, and to achieve that we will use LSC (LDAP Synchronizer Project).
- We will configure LSC to define the password to use SASL
How Do We Implement This?

Configuring OpenLDAP

- We need to specify which schema classes to use
  
  ```
  include /etc/openldap/schema/core.schema
  include /etc/openldap/schema/cosine.schema
  include /etc/openldap/schema/inetorgperson.schema
  ```

- Then we need to define the database
  
  ```
  database hdb
  suffix "dc=cisco,dc=com"
  rootdn "cn=admin,dc=cisco,dc=com"
  rootpw Cisco,123
  ```

- Defining the directory where to store the database
  
  ```
  directory /var/lib/ldap
  ```

- Indices to maintain for this database
  
  ```
  index objectClass eq,pres
  index ou,cn,mail,surname,givenname eq,pres,sub
  index uid eq,pres,sub
  ```
How Do We Implement This?

Configuring OpenLDAP

- Create an LDIF file with the database initiation
  
  ```
  dn: dc=cisco,dc=com
  objectClass: top
  objectClass: dcObject
  objectClass: organization
  dc: cisco
  o: CISCO
  
  dn: OU=People,DC=cisco,DC=com
  objectClass: top
  objectClass: organizationalUnit
  ou: People
  ```

- And import it to OpenLDAP
  
  ```
  ldapadd -x -D cn=admin,dc=cisco,dc=com -w Cisco,123 -f init.ldif -c
  ```
How Do We Implement This?

**Import User into OpenLDAP**

- The conversion of user information from AD to OpenLDAP isn’t a simple task, that are too many differences in the organization of both products.

- So you should use LSC [http://lsc-project.org](http://lsc-project.org)

- Before you install LSC, make sure that you have Java JDK installed in the system.

- LSC will allow you to read the user attributes from the Users created in each AD domain and import them to OpenLDAP

- Apart from adding the users and their attributes to OpenLDAP it will also be responsible to do the update and deletion.

- It should run in sync with the CUCM and other Cisco collaboration products (for example if CUCM synchronize with OpenLDAP at 2.00 AM, LSC should run before than that, his synchronization with AD domains controllers.)
How Do We Implement This?

Configuring LSC

You Will Need to Provide Four Sections in the Configuration File of LSC (If You Install It from rpm, the Configuration File Should Be /etc/lsc/lsc.properties)

- Destination LDAP directory
  
  dst.java.naming.provider.url = ldap://localhost/dc=cisco,dc=com
  
  dst.java.naming.security.authentication = simple
  
  dst.java.naming.security.principal = cn=admin,dc=cisco,dc=com
  
  dst.java.naming.security.credentials = Cisco,123
  
  dst.java.naming.referral = ignore
  
  dst.java.naming.ldap.derefAliases = never
  
  dst.java.naming.factory.initial = com.sun.jndi.ldap.LdapCtxFactory
  
  dst.java.naming.ldap.version = 3

  OpenLDAP Address

  Credentials to Connect
How Do We Implement This?
Configuring LSC

- Destination LDAP directory
  ```
  src.java.naming.provider.url = ldap://ad-cisco-1.cisco.com/dc=cisco,dc=com
  src.java.naming.security.authentication = simple
  src.java.naming.security.principal = cn=Administrator,cn=Users,dc=cisco,dc=com
  src.java.naming.security.credentials = Cisco,123
  src.java.naming.referral = ignore
  src.java.naming.ldap.derefAliases = never
  src.java.naming.factory.initial = com.sun.jndi.ldap.LdapCtxFactory
  src.java.naming.ldap.version = 3
  src.java.naming.ldap.pageSize = 1000
  ```

AD Address

Credentials to Connect
How Do We Implement This?
Configuring LSC

Which task to perform

```java
lsc.tasks.ADuser.srcService = org.lsc.jndi.SimpleJndiSrcService
lsc.tasks.ADuser.srcService.filterAll = (&(objectClass=user)(sAMAccountType=805306368))
lsc.tasks.ADuser.srcService.filterId = (sAMAccountName={sAMAccountName})
lsc.tasks.ADuser.srcService.attrs = lsc.tasks.ADuser.srcService.attrs = dn cn objectClass sn telephoneNumber
  givenName mail userPrincipalName sAMAccountName
lsc.tasks.ADuser.srcService.pivotAttrs = sAMAccountName

lsc.tasks.ADuser.dstService = org.lsc.jndi.SimpleJndiDstService
lsc.tasks.ADuser.dstService.baseDn = ou=People
lsc.tasks.ADuser.dstService.filterAll = (uid=* )
lsc.tasks.ADuser.dstService.filterId = (uid={uid})
lsc.tasks.ADuser.dstService.attrs = cn objectClass sn givenName telephoneNumber mail userPassword uid
lsc.tasks.ADuser.dstService.pivotAttrs = uid

lsc.tasks.ADuser.dn = "CN=" + srcBean.getAttributeValueById("cn") + ",ou=People"
```

Get User from AD

Insert Them in OpenLDAP
How Do We Implement This?

Configuring LSC

- **Syncronization Options**

  \[
  \text{lsc.\text{syncoptions}.ADuser = org.lsc.beans.syncoptions.PropertiesBasedSyncOptions}
  \]

  \[
  \text{lsc.\text{syncoptions}.ADuser.default.action = F}
  \]

  \[
  \text{lsc.\text{syncoptions}.ADuser.objectClass.action = F}
  \]

  \[
  \text{lsc.\text{syncoptions}.ADuser.objectClass.force_value = \{top, person, organizationalperson, inetOrgPerson\}}
  \]

  \[
  \text{lsc.\text{syncoptions}.ADuser.uid.create_value = srcBean.getAttributeValueById("sAMAccountName")}
  \]

  \[
  \text{lsc.\text{syncoptions}.ADuser.userPassword.create_value = \{SASL\} + srcBean.getAttributeValueById("userPrincipalName")}
  \]

  - **Fix ObjectClass “Particularities” of AD**
  - **Covert sAMAccountName to uid**
  - **Create the User Password Attribute to Be Used by SASL**
How Do We Implement This?

Checking the Process

- Run manually LSC
  
  `lsc -f /etc/lsc -s all -c all`

- Check if the users are correctly in OpenLDAP
  
  `ldapsearch -x`

- You users should look like:
  
  ```
dn: CN=Paulo Jorge Correia,ou=People,dc=cisco,dc=com
uid: paucorre
mail: paucorre@cisco.com
sn: Correia
cn: Paulo Jorge Correia
telephoneNumber: 11112
userPassword: {SASL}paucorre@cisco.com
givenName: Paulo
objectclass: person
objectclass: inetOrgPerson
objectclass: organizationalperson
objectclass: top
```
How Do We Implement This?
Configuring SASL

- This method of authentication relay between OpenLDAP and AD is based on the use of SASL (Simple Authentication and Security Layer, defined in RFC 2222). This OpenLDAP to send authentication requests of a user to AD, ie the bind operation of LDAP (through saslauthd), whose role is to contact an Active Directory controller to validate the password provided by the user.

- Install cyrus-sasl-ldap package in you Linux box
How Do We Implement This?

Configuring SASL

Add SASL Authentication to OpenLDAP (/etc/openldap/slapd.conf)
- sasl-host: localhost
- sasl-secprops: none

Configuring init Parameters (/etc/sysconfig/saslauthd)
- SOCKETDIR=/var/run/saslauthd
- MECH=kerberos5
- FLAGS="-O /etc/saslauthd.conf"

Configuring saslauthd (/etc/saslauthd.conf)
- ldap_servers: ldap://localhost
- ldap_search_base: dc=cisco,dc=com
- ldap_timeout: 10
- ldap_filter: uid=%u
- ldap_bind_dn: cn=admin,dc=cisco,dc=com
- ldap_password: Cisco,123
- ldap_deref: never
- ldap_restart: yes
- ldap_scope: sub
- ldap_use_sasl: no
- ldap_start_tls: no
- ldap_version: 3
- ldap_auth_method: bind
How Do We Implement This?

Configuring SASL

Configuring Kerberos 5 (If You Install It from rpm, the Configuration File Should Be /etc/krb5.conf)

```
[realms]
CISCO.COM = {
    kdc = ad-cisco-1.cisco.com
}
EMERG.CISCO.COM = {
    kdc = ad-cisco-2.cisco.com
}
TANDBERG.COM = {
    kdc = ad-tandberg.tandberg.com
}
WEBEX.COM = {
    kdc = ad-webex.webex.com
}
[domain_realm]
  .cisco.com = CISCO.COM
  .emerg.cisco.com = EMERG.CISCO.COM
  .webex.com = WEBEX.COM
  .tandberg.com = TANDBERG.COM
```
Appendix D
OpenAM Installation and Configuration

- Pre-requisite:
  - Active Directory (AD – Domain Controller), Windows clients, CUCM and OpenAM are in the same domain

- Provision OpenAM server in AD and generate keytab files using ktpass command
OpenAM Installation and Configuration

Deploy and Configure OpenSSO Enterprise Using the GUI Configurator
Configure Policies on OpenSSO Server for –

- CUCM User and UDS web app
- Query Parameters
OpenAM Installation and Configuration

Configure J2EE Agent Profile for Policy Agent 3.0
OpenAM Installation and Configuration

Configure Windows Desktop SSO Login Module Instance
OpenAM Installation and Configuration

Configure “Login Form URI” and “OpenAM Login URL” for the PA