Securing Linux using LDAP with z/VM RACF

Alan Altmark
Alan_Altmark@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

DB2
DS8000*
Enterprise Storage Server*
IBM*
IBM eServer
IBM logo*
System Storage*
System z

System z9
System z10
z9*
z10
z/OS*
z/VM
z/VSE
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- What’s the problem?
- How does LDAP help? What is it?
- How LDAP and RACF work together

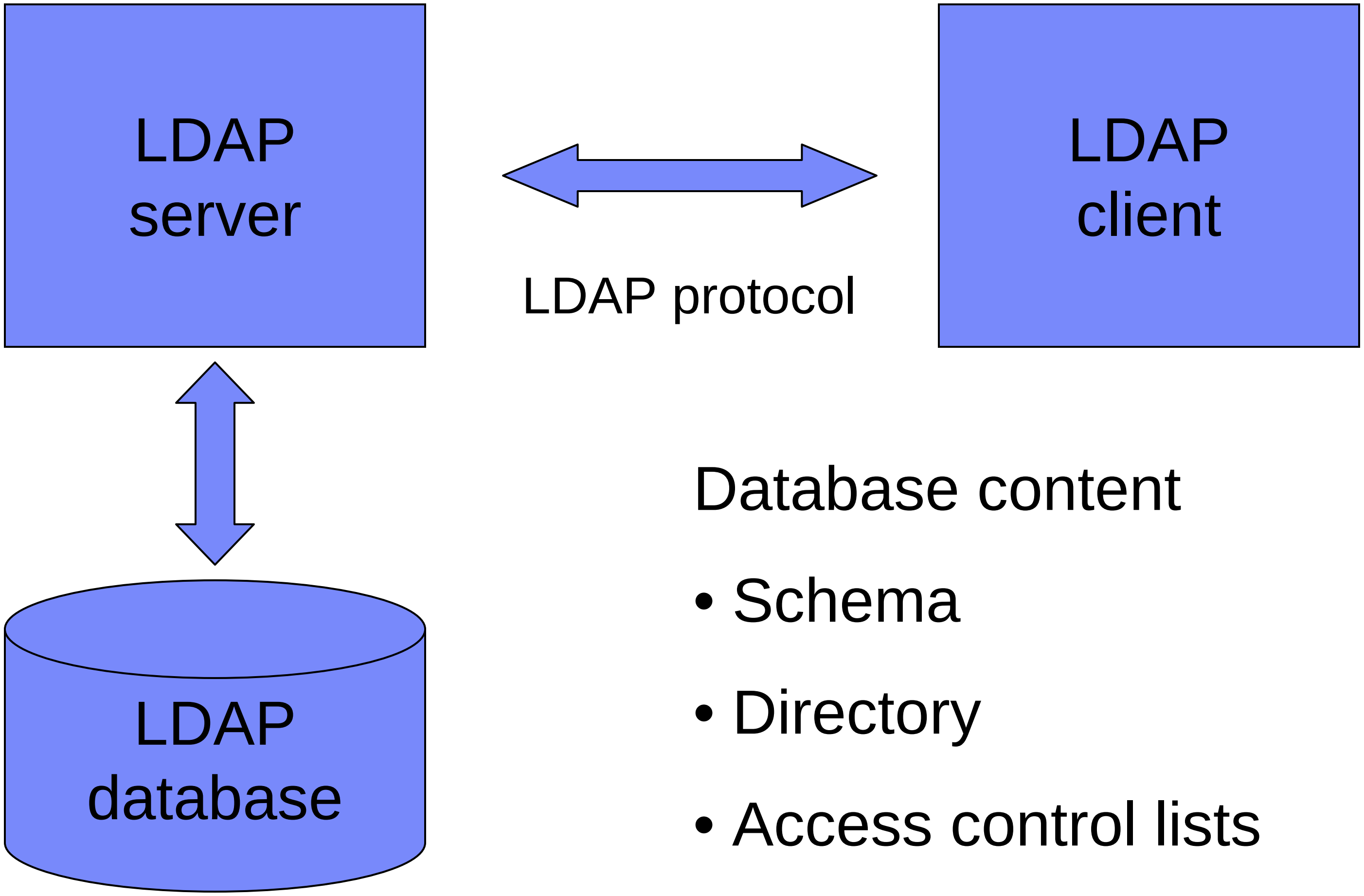
What’s the problem?

- You’ve got lots of Linux servers
- You have the same users on each
- Those users are same users as you have on z/VM
  - flat name space
- You’d like to have a single, central repository for your z/VM *and* Linux passwords
- You’d like that repository to be RACF on z/VM

LDAP

- Lightweight Directory Access Protocol (RFC 2251)
- Standard way for a client to retrieve data stored in a Directory Information Tree (DIT)
- The *schema* defines how the DIT is structured
- *Distinguished name* (DN) identifies a node in the tree
- X.500 model

Conceptual Components



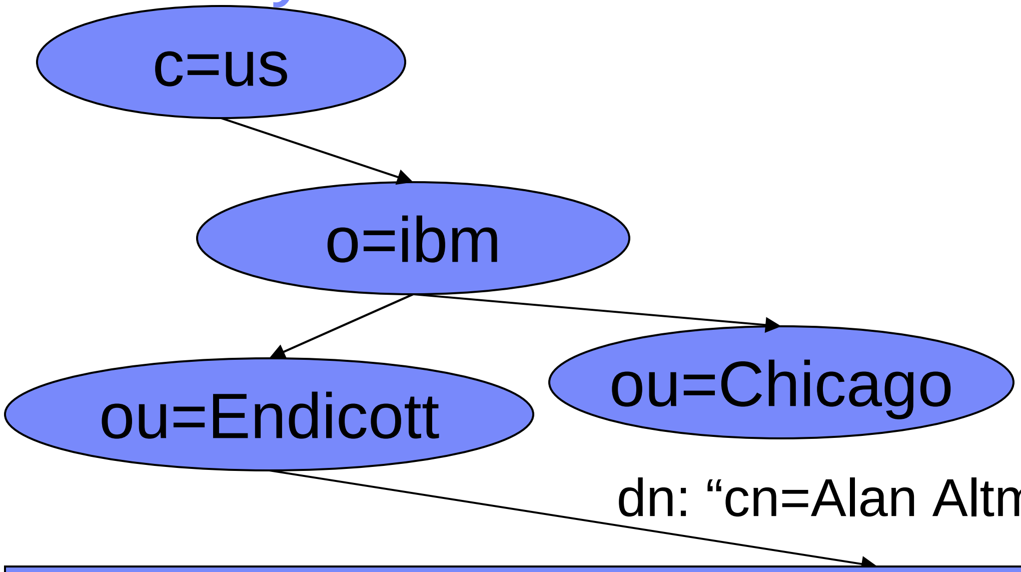
Database content

- Schema
- Directory
- Access control lists

Schema

- A *schema* defines the content of some branch of the Directory Information Tree
  - Object classes
  - Object names
  - Object attributes
    - E.g. data type, instance limit
- The schema can be extended dynamically with the LDAPADD command

Directory


c=us

o=ibm

ou=Endicott

ou=Chicago

c: country

o: organization

ou: organizational unit

cn: common name

dn: distinguished name

dn: “cn=Alan Altmark,ou=Endicott,o=ibm,c=us”

cn: Alan Altmark
phone: 6074293323
addr1: 1701 North Street
city: Endicott
state: NY
zipcode: 13760
empnum: NY123456

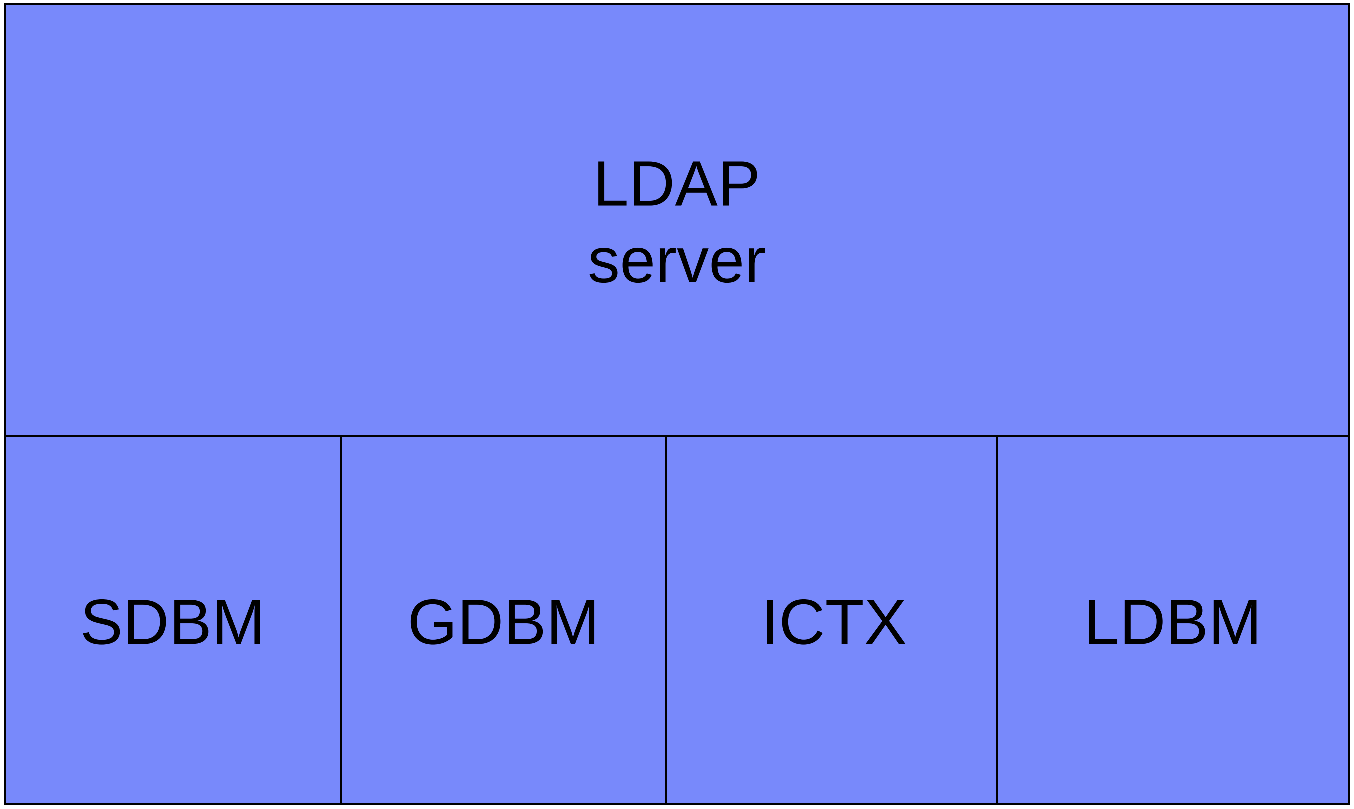
dept: G72G
bldg: 250
floor: 2
office: Y4
uid: aaltmark
ibm-nativeid: ALAN

z/VM LDAP Server

z/VM 5.4

- z/OS 1.10 IBM Tivoli Directory Server (ITDS)
- Each server handles a single Directory Information Tree with a single schema
- Different branches of the tree can be provisioned by different *backends* (database managers)
  - SDBM, LDBM, GDBM, ICTX

z/VM LDAP Server



LDAP
server

SDBM

GDBM

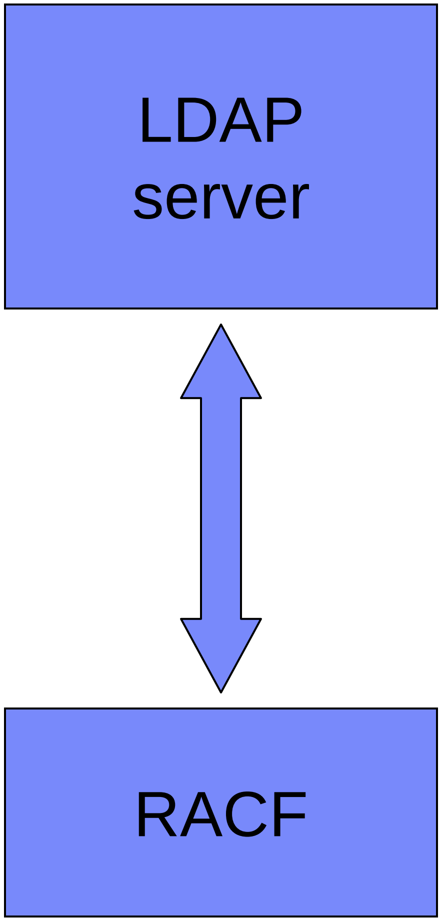
ICTX

LDBM

Server backends

SDBM

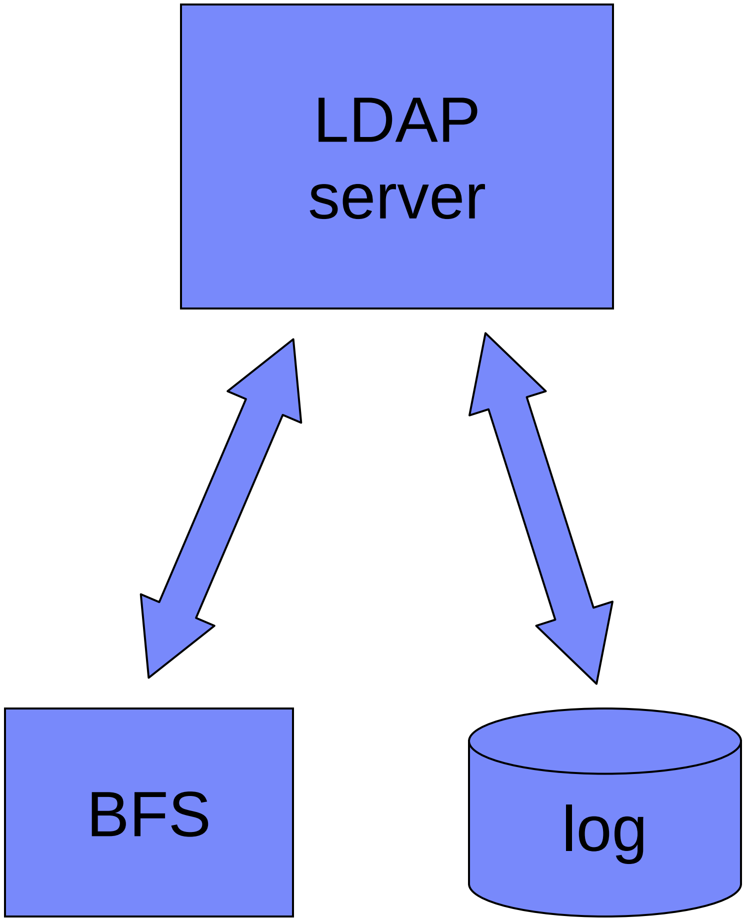
- Uses a RACF-defined schema
- RACF password verification on a bind
- Remote RACF administration
  - Users
  - Groups
  - Connect groups
- RACF only – no other ESMs



GDBM

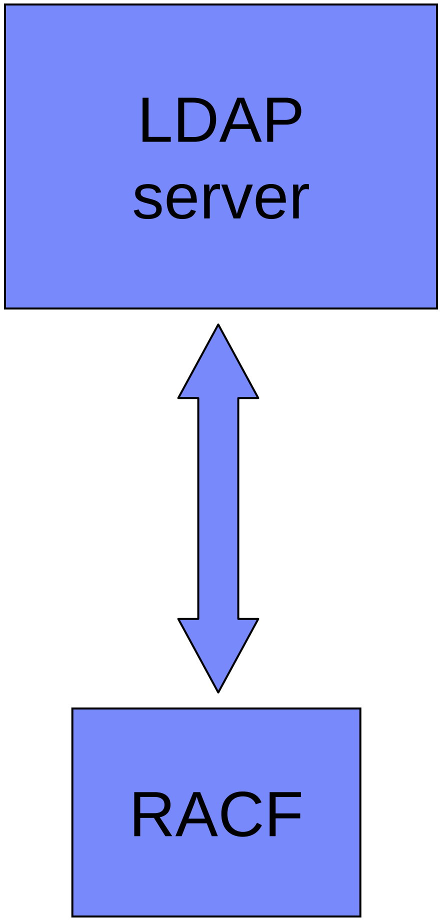
- Logs changes to the LDBM
  - Name of attribute
  - New value of attribute
  - Identity of person who changed it
  - When it was changed

- As of z/VM 5.4, this includes SDBM (RACF).



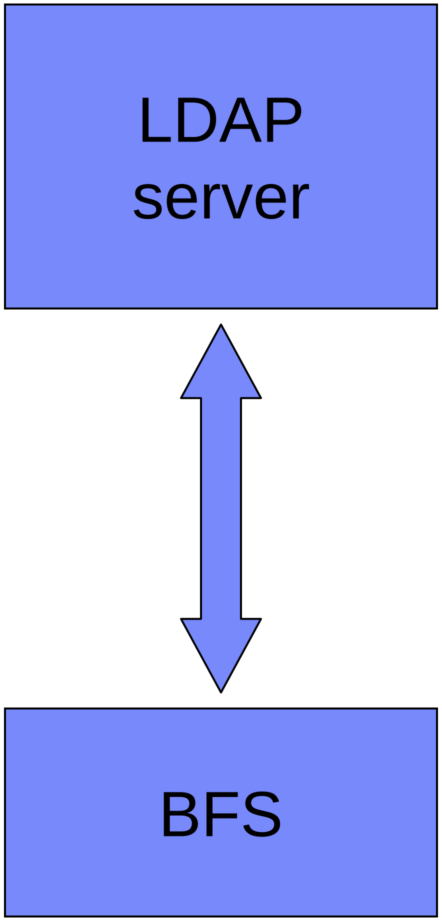
ICTX

- Remote audit and authorization services
  - RACROUTE
- LDAP extended operation (XOP)
- Information in the TCP/IP Programmer's Reference
- Linux audit daemon (auditd) can do this



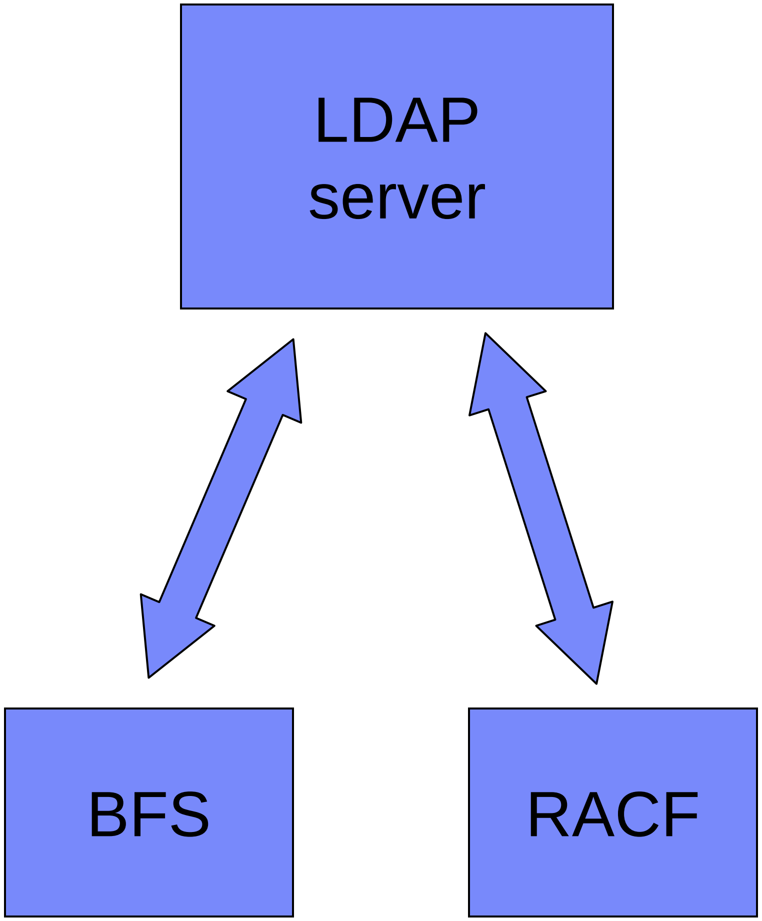
LDBM

- Basic LDAP Database Manager
- Directory is implemented in Byte File System (BFS)
- Full LDAP capability
- Can implement any schema



LDBM with Native Authentication

- LDAP bind authentication performed using RACF
- Full LDAP capability
- uid attribute is used to satisfy LDAP dn search (user lookup)
- RACF user ID is the uid unless ibm-nativeId is present



Logging in from Linux (LDAP-ready PAM)

1. Linux binds to the LDAP server
2. Linux does an LDAP search for *uid* = username
3. LDAP returns a dn (cn=,ou=,o=,c=)
4. Linux does an LDAP bind, handing the LDAP server the dn and the entered password
5. The LDAP server locates the dn and extracts the uid or ibm-nativeid
6. The extracted value and the entered password are given to RACF for verification
7. The LDAP server responds to Linux with an answer of “yes” or “no”

A word about LDAP binds…

- If you do not specify binddn and bindpw in ldap.conf, bind for search will be done anonymously
  - `allowAnonymousBinds on` is required in DS CONF
  - All accesses are as cn=anybody

- Do not use adminDN as bindDN
  - Too much power

- May wish to restrict the data that binddn or cn=anybody can search

LDAP Server Configuration

- DS CONF
  - Everything goes here
- Samples on TCPMAINT 591
  - Excellent commentary
  - LDAP-DS SCONFIG
  - LDAP-DS SAMPENVR
- Production on TCPMAINT 198

DS CONF

`adminDN cn=ldapadm,o=ibm,c=us`	LDAP admin id
`database LDBM GLDBLD31`	Enable LDBM
`suffix o=ibm,c=us`	Default suffix
`useNativeAuth ALL`	Force RACF lookup
`nativeUpdateAllowed YES`	Password change ok
`#useNativeAuth SELECTED`	
`#nativeAuthSubtree ou=Raleigh,o=ibm,c=us`	RACF lookup only
`#nativeAuthSubtree ou=Endicott,o=ibm,c=us`	…on these subtrees

Defining a user to LDAP

- Create an LDIF file that contains the user definition
- Use LDAPADD to store the LDIF data in the LDAP server

LDIF Example

```
dn:  “cn=Alan Altmark,ou=Endicott,o=ibm,c=us”
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: ibm-nativeAuthentication
cn: “Alan Altmark”                         Common name
sn: Altmark                                Surname
uid: aaltmark                              Linux user name
ibm-nativeid: ALTMARKA                     RACF user ID
```

LDIF Example (Referrals)

In corporate LDAP server

```
dn: ou=endicott,o=ibm,c=us
objectclass: referral
objectclass: extensibleObject
ref: ldap://ldap.endicott.ibm.com/ou=endicott,o=ibm,c=us
```

In local Endicott server

```
referral ldap://ldap.ibm.com
adminDN cn=ldapadm,ou=endicott,o=ibm,c=us
database ldbm GLDBLD31
suffix ou=endicott,o=ibm,c=us
```

LDIF Example

- Issue LDAPADD command from CMS

- ldapadd –h loopback –D “cn=ldapadm”
  –w *password* –f //filename.filetype

Name Information Service (NIS)

- Enables retrieval of user configuration data from remote LDAP server using Name Service Switch (NSS)

- RFC 2307

- No entry in etc/passwd, etc/shadow, or etc/groups

- Download NIS schema from ftp://www.redbooks.ibm.com/redbooks/REDP0221
  - It adds the POSIX information to a user's LDAP entry

- Details in *Security on z/VM* from IBM Redbooks

Secure LDAP connections

- SSL/TLS may be optionally used by both the z/VM LDAP clients and server
  - All secure binds should be encrypted
- For the clients, certificate management is provided by an SSL/TLS stack (“CMS System SSL”) that runs in the user virtual machine
  - Does not use the SSL server
- The LDAP server can use the SSL server or can use CMS System SSL directly

System SSL

- A set of utilities to manage the X.509 certificates that can be used by the LDAP client utilities and server for authentication and encryption
- CMS, not Linux
  - This is the basis for the z/VM 5.4 SSL server
- Uses BFS and the POSIX shell
- gskkyman
  - command line interface
  - menu

Certificate Mangement

- Handles certificate renewals
- Export and import of certificate and private key
  - Enables easy sharing of certificates
- Be your own Certificate Authority (CA)

References

- Redbooks
  - Understanding LDAP: Design and Implementation, SG24-4986
  - Securing Linux on zSeries with a Central z/OS LDAP Server, REDP-0221
  - Advanced LDAP User Authentication, REDP-3863
  - Security on z/VM, SG24-7471

- z/VM TCP/IP Planning and Customization
  - SC24-6124

- z/VM TCP/IP LDAP Administration Guide
  - SC24-6140

Thanks for listening!

www.VM.ibm.com/devpages/altmarka

e-mail: Alan_Altmark@us.ibm.com