

# LDAP Directories

Matthew Geddes  
mgeddes@augment-it.com.au

17th May 2005

# Part Outline

- 1 Outline
- 2 What is LDAP?
  - What is LDAP?
  - Why use LDAP?
  - LDAP implementations
  - Common applications
- 3 LDAP concepts
  - Entries, objects and attributes
  - Distinguished names
  - Filter strings
  - LDIF format

# What is LDAP?

- Lightweight Directory Access Protocol
- Based on X.500 Directory Access Protocol
- Tree structured
- Often heavily read-optimised
- Pretty cool

# What is LDAP?

- Lightweight Directory Access Protocol
- Based on X.500 Directory Access Protocol
- Tree structured
- Often heavily read-optimised
- Pretty cool

# What is LDAP?

- Lightweight Directory Access Protocol
- Based on X.500 Directory Access Protocol
- Tree structured
- Often heavily read-optimised
- Pretty cool

# What is LDAP?

- Lightweight Directory Access Protocol
- Based on X.500 Directory Access Protocol
- Tree structured
- Often heavily read-optimised
- Pretty cool

# What is LDAP?

- Lightweight Directory Access Protocol
- Based on X.500 Directory Access Protocol
- Tree structured
- Often heavily read-optimised
- Pretty cool

# Why use LDAP?

Need for:

- Structured data
- Standards-compliant network protocol
- More reads than writes
- Simple records

# Why use LDAP?

Need for:

- Structured data
- Standards-compliant network protocol
- More reads than writes
- Simple records

# Why use LDAP?

Need for:

- Structured data
- Standards-compliant network protocol
- More reads than writes
- Simple records

# Why use LDAP?

Need for:

- Structured data
- Standards-compliant network protocol
- More reads than writes
- Simple records

# LDAP implementations

- OpenLDAP
- Apple's OpenDirectory (OpenLDAP rebadged)
- Novell eDirectory
- IBM's LDAP directory
- Sun/Netscape directory (ex-iPlanet)
- Microsoft Active Directory

# Common applications of LDAP directories

- Contacts databases, including MTA and mail client lookups
- User accounts databases, including Unix (RFC2307), Samba and Kerberos
- Equipment register / name resolution
- Distributed name service switch directory. Thin clients, server farms, etc

# Common applications of LDAP directories

- Contacts databases, including MTA and mail client lookups
- User accounts databases, including Unix (RFC2307), Samba and Kerberos
- Equipment register / name resolution
- Distributed name service switch directory. Thin clients, server farms, etc

# Common applications of LDAP directories

- Contacts databases, including MTA and mail client lookups
- User accounts databases, including Unix (RFC2307), Samba and Kerberos
- Equipment register / name resolution
- Distributed name service switch directory. Thin clients, server farms, etc

# Common applications of LDAP directories

- Contacts databases, including MTA and mail client lookups
- User accounts databases, including Unix (RFC2307), Samba and Kerberos
- Equipment register / name resolution
- Distributed name service switch directory. Thin clients, server farms, etc

# Objects and attributes

- Entry - collection of attributes belonging to objects
- Object - a group of attributes
- Attribute - small piece of data

```
dn: dc=LinuxSA,dc=org,c=AU  
dc: LinuxSA  
objectClass: dcObject
```

## Distinguished names

- Each entry in an LDAP directory must have a DN attribute
- The DN describes the object's location in the LDAP directory
- The DN is a similar concept to the path name of a file
- DNs use a ',' as the separator
- DNs have the root of the directory on the *right*-hand side
- DN example:  
`cn=Matthew Geddes,ou=People,dc=LinuxSA,dc=org,c=AU`

## Distinguished names

- Each entry in an LDAP directory must have a DN attribute
- The DN describes the object's location in the LDAP directory
- The DN is a similar concept to the path name of a file
- DNs use a ',' as the separator
- DNs have the root of the directory on the *right*-hand side
- DN example:  
`cn=Matthew Geddes,ou=People,dc=LinuxSA,dc=org,c=AU`

## Distinguished names

- Each entry in an LDAP directory must have a DN attribute
- The DN describes the object's location in the LDAP directory
- The DN is a similar concept to the path name of a file
- DNs use a ',' as the separator
- DNs have the root of the directory on the *right*-hand side
- DN example:  
`cn=Matthew Geddes,ou=People,dc=LinuxSA,dc=org,c=AU`

## Distinguished names

- Each entry in an LDAP directory must have a DN attribute
- The DN describes the object's location in the LDAP directory
- The DN is a similar concept to the path name of a file
- DNs use a ',' as the separator
- DNs have the root of the directory on the *right*-hand side
- DN example:  
`cn=Matthew Geddes,ou=People,dc=LinuxSA,dc=org,c=AU`

## Distinguished names

- Each entry in an LDAP directory must have a DN attribute
- The DN describes the object's location in the LDAP directory
- The DN is a similar concept to the path name of a file
- DNs use a ',' as the separator
- DNs have the root of the directory on the *right*-hand side
- DN example:  
`cn=Matthew Geddes,ou=People,dc=LinuxSA,dc=org,c=AU`

# Filter strings

- Simple: `'cn=Matthew Geddes', 'objectClass=*`
- Complex: `'(&(cn=Matthew Geddes)  
(objectClass=posixAccount))'`

*Note: The single quote characters are only used to escape the filter string in most shells.*

## Filter strings

- Simple: `'cn=Matthew Geddes', 'objectClass=*`
- Complex: `'(&(cn=Matthew Geddes)  
(objectClass=posixAccount))'`

*Note: The single quote characters are only used to escape the filter string in most shells.*

# LDIF format

- Simple series of key/value pairs that make up objects
- Supported by most LDAP implementations

```
dn: cn=Matthew Geddes,ou=People,dc=linuxsa,dc=org,c=au
objectclass: person
cn: Matthew Geddes
sn: Geddes
userpassword: {SSHA}psiaRh4CyVnkr0CJ9mcoVsJbxUdjqmXd
telephoneNumber: 0402 474 232
description: Example user account
```

# Part Outline

- 4 Outline
  
- 5 Architecture
  - Overview
  - Note to self

# Typical LDAP architecture

- 1 LDAP clients - PAM module, MTAs, mail clients, etc
- 2 LDAP protocol
- 3 LDAP service(s)
- 4 Data store - often BerkeleyDB-style database

## Note to self...

Draw pretty picture on whiteboard and hope that it draws attention away from your incompetence.

# Part Outline

- 6 Outline
  - slapd.conf
  - Populating the directory
  - LDAP schemas

# Hash administrator password

The `slappasswd` tool can hash a password for inclusion in a configuration file:

```
matthew@angus:matthew \$ slappasswd  
New password:  
Re-enter new password:  
{SSHA}j7rD1PLMQL8mD1D4MM4XD+qNpo0jmv96
```

# Add directory to slapd.conf

Add new entry to `/etc/openldap/slapd.conf` similar to:

```
database      ldbm
suffix        "dc=linuxsa,dc=org,c=AU"
directory     /var/db/openldap/openldap-data/linuxsa.org.au/
index         uid,cn,mail eq
rootdn        "cn=admin,dc=linuxsa,dc=org,c=AU"
rootpw        {SSHA}j7rD1PLMQL8mD1D4MM4XD+qNpo0jmv96
```

# Restart slapd service

- 1 Check the syntax of the config file using slapd's `-t` option
- 2 Restart the slapd service either manually, or using a SysV-style init script:

```
matthew@angus:LDAP \ $ sudo /usr/libexec/slapd
```

or

```
matthew@angus:LDAP \ $ sudo /etc/init.d/ldap restart
```

# Steps to populating a directory

- 1 Create an LDIF file containing all of the entries you want created
- 2 Use either `slapadd` or `ldapadd` to add the entries
- 3 Use `ldapsearch` to do a test query

# Sample LDIF file

```
dn: dc=LinuxSA,dc=org,c=AU
dc: LinuxSA
objectClass: domainComponent

dn: ou=People,dc=LinuxSA,dc=org,c=AU
objectClass: organizationalUnit
ou: People

dn: cn=Matthew Geddes,ou=People,dc=linuxsa,dc=org,c=au
objectclass: person
cn: Matthew Geddes
sn: Geddes
userpassword: {SSHA}psiaRh4CyVnkr0CJ9mcoVsJbxUdjqmXd
telephoneNumber: 0402 474 232
description: Example user account
```

# Add entries to the LDAP directory

```
matthew@angus:LDAP \ $ ldapadd -D cn=admin,dc=LinuxSA,dc=org,c=AU \  
-W -x < ./sample.ldif  
Enter LDAP Password:  
adding new entry "dc=LinuxSA,dc=org,c=AU"  
  
adding new entry "ou=People,dc=LinuxSA,dc=org,c=AU"  
  
adding new entry "cn=Matthew Geddes,ou=People,dc=linuxsa,dc=org,c=au"
```

# Test search

```
matthew@angus:LDAP \ $ ldapsearch -D \  
    "cn=Matthew Geddes,ou=people,dc=linuxsa,dc=org,c=au" \  
    -b dc=linuxsa,dc=org,c=au -w crud -x -s sub \  
    'sn~=Gedes'  
...  
# Matthew Geddes, People, linuxsa, org, au  
dn: cn=Matthew Geddes,ou=People,dc=linuxsa,dc=org,c=au  
objectClass: person  
cn: Matthew Geddes  
sn: Geddes  
userPassword:: e1NTSEF9cHNpYVJoNEN5Vm5rck9DSjltY29Wc0pieFVkanFtWGQ=  
telephoneNumber: 0402 474 232  
description: Example user account  
...
```

# LDAP schemas

```
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'  
  DESC 'An integer uniquely identifying a user'  
  EQUALITY integerMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )  
  
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY  
  DESC 'Abstraction of an account with POSIX attributes'  
  MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )  
  MAY ( userPassword $ loginShell $ gecos $ description ) )
```

# Part Outline

## 7 Outline

## 8 Additional user attributes

- posixAccount attributes
- inetOrgPerson attributes
- Samba accounts for Windows users
- Mail routing account

## Unix authentication (RFC2307)

- Configure PAM and NSS LDAP modules (<http://www.padl.com/>)
- Add posixAccount attributes to entry:

```
matthew@angus:LDAP $ ldapmodify -D cn=admin,dc=linuxsa,dc=org,c=AU \  
-w linuxsa -x < ./posixAccount.ldif  
  
modifying entry "cn=Matthew Geddes, ou=People, dc=LinuxSA, dc=org, c=AU"  
  
matthew@angus:LDAP \$ cat posixAccount.ldif  
dn: cn=Matthew Geddes, ou=People, dc=LinuxSA, dc=org, c=AU  
objectClass: posixAccount  
uid: mgeddes  
uidNumber: 1000  
gidNumber: 1000  
homeDirectory: /home/mgeddes  
loginShell: /bin/bash  
gecos: Matthew Geddes
```

# inetOrgPerson attributes

```
matthew@angus:LDAP $ ldapmodify -D cn=admin,dc=linuxsa,dc=org,c=AU \  
    -w linuxsa -x < ./inetOrgPerson.ldif  
modifying entry "cn=Matthew Geddes, ou=People, dc=linuxsa, dc=org, c=AU"  
  
matthew@angus:LDAP $ cat inetOrgPerson.ldif  
dn: cn=Matthew Geddes, ou=People, dc=linuxsa, dc=org, c=AU  
objectClass: inetOrgPerson  
mail: Matthew.Geddes@linuxsa.org.au  
jpegPhoto:< file:///tmp/mattgeddes.jpeg  
mobile: 0402 474 232
```

## Samba accounts for Windows users

```
matthew@angus:LDAP $ ldapmodify -D cn=admin,dc=linuxsa,dc=org,c=AU \  
-w linuxsa -x < ./sambaAccount.ldif  
modifying entry "cn=Matthew Geddes, ou=People, dc=LinuxSA, dc=org, c=AU"  
  
matthew@angus:LDAP $ cat sambaAccount.ldif  
dn: cn=Matthew Geddes, ou=People, dc=LinuxSA, dc=org, c=AU  
rid: 503  
lmPassword: 6166F82AFC46DBBFAAD3B435B51404EE  
ntPassword: D28AFE257D9F0090AC6C0EC11681D5C1  
smbHome: \\sambapdc\mgeddes  
homeDrive: H:  
scriptPath: login.bat  
domain: LINUXSA  
acctFlags: [U          ]
```

# Mail routing account

```
matthew@angus:LDAP $ ldapmodify -D cn=admin,dc=linuxsa,dc=org,c=AU \  
    -w linuxsa -x < ./inetLocalMailRecipient.ldif  
modifying entry "cn=Matthew Geddes, ou=People, dc=linuxsa, dc=org, c=AU"  
  
matthew@angus:LDAP $ cat inetLocalMailRecipient.ldif  
dn: cn=Matthew Geddes, ou=People, dc=linuxsa, dc=org, c=AU  
objectClass: inetLocalMailRecipient  
mailLocalAddress: mgeddes@mail5.linuxsa.org.au  
mailRoutingAddress: Matthew.Geddes@linuxsa.org.au
```

# Part Outline

- 9 Outline
  
- 10 Simple queries in Python
  - Importing the ldap module
  - Connect and bind
  - Performing a search
  - Processing results
  - Resulting output

# Importing the ldap module

The python-ldap module is available from  
<http://python-ldap.sf.net>

```
import ldap
```

# Connecting and binding

```
binddn = "cn=Matthew Geddes,ou=People,dc=LinuxSA,dc=org,c=AU"  
bindpw = "crud"  
ldapservers = "localhost"  
  
ldap_connection = ldap.open(ldapservers)  
ldap_connection.simple_bind_s(binddn, bindpw)
```

# Performing a search

```
basedn = "dc=LinuxSA, dc=org, c=AU"  
filter = "cn~=Matt"  
  
results = ldap_connection.search_s(basedn, ldap.SCOPE_SUBTREE, filter)
```

## Processing search results

```
num_entries = len(results)
print "# Returned %d entries" % num_entries

for entry in results:
    dn = entry[0]
    print "dn: %s" % dn
    for attribute in entry[1]:
        for value in entry[1][attribute]:
            if attribute != 'jpegPhoto':
                print "%s: %s" % (attribute, value)
    print ""
```

## Resulting output

```
# Returned 1 entries
dn: cn=Matthew Geddes,ou=People,dc=linuxsa,dc=org,c=au
domain: LINUXSA
cn: Matthew Geddes
objectClass: inetOrgPerson
uidNumber: 1000
rid: 503
mailRoutingAddress: Matthew.Geddes@linuxsa.org.au
uid: mgeddes
userPassword: {SSHA}psiaRh4CyVnkr0CJ9mcoVsJbxUdjqmXd
scriptPath: login.bat
mail: Matthew.Geddes@linuxsa.org.au
acctFlags: [U          ]
lmPassword: 6166F82AFC46DBBFAAD3B435B51404EE
description: Example user account
loginShell: /bin/bash
gidNumber: 1000
```

## ... continued

```
telephoneNumber: 0402 474 232  
homeDrive: H:  
ntPassword: D28AFE257D9F0090AC6C0EC11681D5C1  
mobile: 0402 474 232  
smbHome: \\sambapdc\mgeddes  
gecos: Matthew Geddes  
sn: Geddes  
homeDirectory: /home/mgeddes  
mailLocalAddress: mgeddes@mail5.linuxsa.org.au
```

## Further reading

- <http://www.openldap.org/> - OpenLDAP project
- <http://www.padl.com/> - Luke Howard's PADL project
- Understanding and Deploying LDAP Directory Services - The LDAP bible
- The O'Reilly-published LDAP book
- <http://www.tldp.org/> - Various LDAP-related HOWTOs

## Miscellaneous cruft

- These slides were created using  $\text{\LaTeX}$  and the beamer slideshow package to create a PDF
- Any questions?