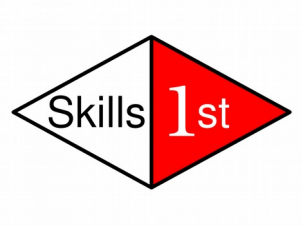

LDAP Basics: Exercises

Version 0.1

22nd November 2015

Andrew Findlay

Skills 1st Ltd

www.skills-1st.co.uk

Dr Andrew Findlay
Skills 1st Ltd
2 Cedar Chase
Taplow
Maidenhead
SL6 0EU
01628 782565
andrew.findlay@skills-1st.co.uk

Working Environment

You have a virtual machine in the Amazon AWS cloud, running CentOS 7. The Amazon cloud uses private address space internally, so although you will connect using a public IP address you may need to be aware of the machine's local address on the 172.30.* network.

We use VNC to connect to virtual screens on the servers. Most Linux distros provide *vncviewer* for this purpose. If your local machine uses a different operating system you may need to download a VNC client program from the net.

The VNC password is *skills782565*.

You can also connect with SSH: login as *student* – password *skills782565*

When you need to do something as root, use *su* (the password is also *skills782565*)

OpenLDAP

CentOS supplies OpenLDAP packages, but the only one loaded on these machines is the library package needed by other OS components.

OpenLDAP has been built from source and installed in `/usr/local`

You will find the source code and config options in `/home/student/src`

1 First LDAP Server

In this exercise you will configure OpenLDAP and load it with some sample entries.

1.1 Basic LDAP server setup

Most of the exercises will run as *student* – we do not need to use *root* and it is safer not to do so. At times we will be running several LDAP servers at once: each will listen on a different TCP port and will have its own database.

1. Move to the directory containing the first server files:

```
cd ~/exercises/first-server
```

2. Inspect the `data.ldif` file.

3. Load the LDAP data:

```
slapadd -f slapd.conf -l data.ldif
```

4. Start the server:

```
slapd -h ldap://:1389/ -f slapd.conf
```

   No messages are expected here. The server will start in the background and listen on the URL specified. In this case, `ldap://:1389/` means 'listen on TCP port 1389 on all interfaces'. Slapd will listen on IPv4 and also IPv6 if both are available.

5. Now try a simple command-line search:

```
ldapsearch -x -H ldap://localhost:1389/ \
    -b dc=example,dc=org uid=tom
```

   Note the use of capital H to introduce the LDAP URL here.
   The search base is specified with the *-b* option.
   The search filter in this case is very simple: *uid=tom*
   The *-x* flag is necessary on most systems to disable SASL.
   You should get a single result from this search.

6. Try another search, but this time replace *uid=tom* with *mail=nic@example.org*
   You should get a single entry again.

7. To avoid typing the LDAP URI and search base each time you can put them in a file called `ldaprc` in the current directory:

```
URI ldap://ldap1.example.org:1389/
BASE dc=example,dc=org
```

   The LDAP tools also read `~/.ldaprc ~/ldaprc` and several other files. You can override the files using command-line options.

8. Try more searches:

```
ldapsearch -x uid=nosuchuser
ldapsearch -x 'mail=*@example.org'
```

Note that finding no results is not an error.
The wildcard search has to be quoted to protect it from the shell's globbing mechanism.

9. If you want to read a single entry then use a 'base' search:

```
ldapsearch -x -b dc=people,dc=example,dc=org -s base
```

10. The LDAP server provides certain information about itself in the Root DSE (the entry with no name). Use ldapsearch to inspect it:

```
ldapsearch -x -b '' -s base 'objectclass=*'
```

Note minimal data is shown – there are very few 'user' attributes here.

```
ldapsearch -x -b '' -s base 'objectclass=*' +
```

Now we see useful data, particularly *namingContexts* and *supportedSASLMechanisms*. The '+' parameter requests all the operational attributes. If we want user attributes too we can also ask for '*'. Some older LDAP servers do not support the '+' notation: in such cases you have to list each operational attribute that you want to read.

11. The LDAP server has a 'root' user which is all-powerful just like the Linux one. We can invoke that power by adding command-line options to LDAP commands:

```
ldapsearch -x -D cn=root,dc=example,dc=org -w secret \
   uid=tom
```

The name of the root user may vary from one server to another.
Note the password *(secret)* on the command line – this is bad for security as it shows up in *ps* listings and also in the command history. Better security is achievable, e.g. using the *-W* or *-y* flag instead of *-w*.

Look at the search result. There is one more attribute that we have not seen before: *userPassword*. It looks encrypted, but in fact it is just encoded in Base64. Paste the value into this command and see what you get:

```
base64 -d; echo
```

LDAP servers provide various ways to improve password protection, which you should always use on production systems.

2 Apache Directory Studio

This is a graphical LDAP browser/editor based on the Eclipse development environment. If you have Java on your laptop you can download Studio from the Apache website and install it locally: this will give the best performance but the download is over 100MB so it could take some time. The instructions here assume that you will use Studio on the remote machine.

2.1 Setup

1. Connect using VNC, open a terminal window and run Studio:

```
studio
```

2. When the interface starts up, open the workbench, go to the bottom-left panel and create a new LDAP connection with these characteristics:

   Name: ldap1 1389 Anonymous
   Hostname: ldap1.example.org
   Port: 1389
   Authentication method: No authentication

   Leave all other parameters at default settings.

2.2 Browse the DIT

1. You should now be able to browse the LDAP tree. Find the uid=tom entry and double-click on the mail attribute to change the value. The server should refuse to accept the change.
2. Try a search: select a starting point by clicking on an entry in the tree browser. Above the browser panel is a row of search parameters: hover the mouse over each to find out what it does, then set up a subtree search for *uid=nic*

2.3 Browse Schema

Apache Directory Studio has a very useful schema browser.

1. Right-click on one of your LDAP connections in the bottom-left panel and select *Open Schema Browser*
2. The browser opens in a tab in the main panel. Along the bottom it has tabs of its own: select *Object Classes*
3. Find *inetOrgPerson* and select it. On the right you will see information about the class.
4. Expand the lists of *Must* and *May* attributes.

5. Click on *cn*. You are now looking at the Attribute Types display. On the right you see a description of the attribute, along with the list of matching rules that are valid when searching it.

All of the schema information is retrieved from the LDAP server, so it may vary from one server to another. If you have several LDAP connections open you should be careful to check which server's schema you are looking at: there is a *Browse* button top-right to change to another server.

It is not possible to change schema though this browser, though many LDAP servers do permit schema changes via LDAP.

3 Simple data management

3.1 Add a new entry from file

1. The file me.ldif contains a skeleton *person* entry.
2. Edit the file and change the entry to describe yourself. Make sure that the DN is consistent with the attributes in the entry (*uid* in this case)
3. Load the file:

```
ldapadd -x -D cn=root,dc=example,dc=org -w secret \
        -f me.ldif
```

4. Now check that you can find your new entry using Apache Directory Studio. You may need to use the *refresh* button.

3.2 Modify data with Directory Studio

1. Create another LDAP connection:

Name:	ldap1 1389 Root
Hostname:	ldap1.example.org
Port:	1389
Authentication method:	Simple
Bind DN:	cn=root,dc=example,dc=org
Bind Password:	secret

   Leave all other parameters at default settings.
2. Note how each connection is kept separate along with the work done using it.
3. Use the Root connection to modify your new entry.
4. Use the *ldapsearch* command-line tool to verify your changes.

3.3 Non-ASCII Characters

Most attribute types in LDAP use the UTF-8 character set so they can represent glyphs from almost every human language.

1. Use Directory Studio to modify the DisplayName attribute of a person entry to contain some non-ASCII characters. For example, you can type umlauts using AltGr-{ followed by a vowel character. AltGr-4 should generate the Euro symbol.
2. Read the entry using both Directory Studio and *ldapsearch*. You will find that *ldapsearch* now shows the value encoded in BASE64: this can be decoded using the command:

```
base64 -d; echo
```

3. Try adding non-ASCII characters to the *gecos* attribute. What happens and why?

4 Authentication and Authorisation

In this exercise we will use the command-line tools to walk through the process of authentication. We will then verify group membership to determine the authorisation granted to the user.

1. Make sure that your server is running.
2. Remember that we set the URI and search base in ldaprc earlier, so these parameters are still in effect. The person with username 'roger' wants to login, so we start with a subtree search for *uid=roger* in a suitable type of entry:

```
ldapsearch -x '(&(uid=roger)(objectclass=inetorgperson))'
```

   In a real application, the search would include a list of the attributes that we want. There is no point in retrieving JPEG photos at this stage!

3. If that search does not return exactly one entry then we know that the username is invalid. If we do get one entry then we use its DN along with the password supplied by the user to attempt an LDAP bind. To simulate that we will do another search, first using a bad password and then using a good one:

```
ldapsearch -x -D uid=roger,dc=people,dc=example,dc=org \
    -w bad '(uid=roger)'
ldapsearch -x -D uid=roger,dc=people,dc=example,dc=org \
    -w notverysecret '(uid=roger)'
```

4. We have now validated the username and password. The next stage is to check group membership for authorisation. There are several ways to do this, depending on whether we want to check a specific group or to find the list of all groups that the user is a member of.

   Here is one way to check for membership of the 'manager' group.

```
ldapsearch -x '(&(objectclass=groupofnames)(cn=manager)
    (member=uid=roger,dc=people,dc=example,dc=org))'
```

   The search succeeds but it returns no entries: roger is not a member of this group.

5. Try the same search for 'tom' in place of 'roger'.

   Look at the entry that you found. It has a lot of members listed. In some organisations there are groups with millions of members: clearly we do not want to retrieve all that useless data when all we want to know is whether Tom is a member of the group.

6. Repeat the 'tom' and 'roger' searches but this time just request the *cn* attribute to be returned. This is much more efficient:

```
ldapsearch -x '(&(objectclass=groupofnames)(cn=manager)
    (member=uid=tom,dc=people,dc=example,dc=org))' cn
```

7. Now suppose that we want a list of all groups that Tom is a member of. The search is simple:

```
ldapsearch -x '(&(objectclass=groupofnames)
    (member=uid=tom,dc=people,dc=example,dc=org))' cn
```

8. Another sort of group is used in RFC2307 to model the Unix group file. This uses *memberUID* in place of *member* so the search is slightly different. Here is a search to check whether Tom is a member of the 'dialout' group:

```
ldapsearch -x '(&(objectclass=posixgroup)(cn=dialout)
    (memberuid=tom))' cn
```

5 Advanced Topics

If you have time, you might like to try using TLS to secure the LDAP sessions. There are some pre-built certificates in the `exercises/x509` directory, and a ready-made server configuration in `exercises/tls`.

Make sure you stop your existing server before trying to start the new one (or use a different TCP port number for it).

Note the contents of the `ldaprc` file: you will find config to enable TLS, but this does not force it to be used. You can do that by adding the `-ZZ` parameter to *ldapsearch*.

Apache Directory Studio can also use TLS. You will need to add the CA certificate to its certificate store.