10 ACLs and Limits

10.1 Protect person entries

In this exercise we will protect entries representing people, by making them visible only to authenticated users. Anonymous users will still be able to see other types of entry.

1. Prepare the new LDAP server:

```
killall slapd
cd ~/exercises/acls-1
slapadd -f slapd.conf -l data.ldif
slapd -f slapd.conf -h ldap://:1389/
```

2. Test the current behaviour:

```
ldapsearch -x uid=tom
ldapsearch -x -D uid=fiona,dc=people,dc=example,dc=org \
           -w notverysecret uid=tom
```

   Both searches should return the same data.

3. Edit slapd.conf and add this rule to the main database ACL (which is right at the end of the file). Place the new rule above the final 'access to * by * read' rule so that it takes precedence.

```
access to filter="(objectclass=person)"
       by users read
       by * none
```

   The rule is triggered by a match on objectclass, so it will only apply to entries of class *person*. Due to class inheritance, this includes *organizationalPerson* and *inetOrgPerson*.

4. Restart the server and test again with *ldapsearch*. This time the anonymous search should return zero results and the authenticated search should return one result exactly as before.

5. Check that the restriction does not apply to other types of entry:

```
ldapsearch -x '(objectclass=*)'
```

   You should see all the non-person entries.

10.2 Protect specific attributes

In this exercise we will limit the visibility of telephone numbers to members of the *managers* group.

1. Edit slapd.conf again and add a new rule just above the one that you added in exercise 10.1:

```
access to attrs="telephoneNumber"
       by group="cn=manager,dc=groups,dc=example,dc=org" read
       by * none
```

This must come before the more general rule that we used in exercise 10.1 because that rule applies to all attributes in the entry, and if it were executed first then our new rule would never be used.

2. Restart the server and test. Try the search with each of these users:

```
uid=fiona,dc=people,dc=example,dc=org
uid=nb,dc=people,dc=example,dc=org
```

The *uid=fiona* account is not in the managers group, so it should not see any telephone numbers. The *uid=nb* account is in the managers group so it should see telephone numbers.

You should also check that anonymous searches still cannot see any *person* entries.

3. Add a rule that permits authenticated users to search the *uid* attribute but not to read the values. Restart the server and test.

Note that even though the ACL now prevents the *uid* attribute from being returned in the search result, the data is still exposed because it is part of the entry DN. This is one reason why it is best to avoid using meaningful attributes in the DN.

4. Modify your ACL so that anonymous users may see entries for people whose surnames begin with the letter B. Restart the server and test.

10.3 Limits

We use size limits to protect the service from searches that would otherwise return excessively large results. We can also use them to make it harder for malicious users to harvest large amounts of data. Limits can be applied globally or separately for each backend database: in this exercise we will apply them to the main database as that gives the most detailed control.

1. Edit slapd.conf and add limits rules to the main database section: place them just above the access-control list.

```
limits anonymous size.soft=2 size.hard=2
limits users size.soft=10 size.hard=100
```

The effect is to limit anonymous users to just 2 results from each search operation. Authenticated users will get up to 10 results by default, but can ask for a higher limit of up to 100 results.

2. Restart the server.
3. Try these test searches:

```
ldapsearch -x dc=people
ldapsearch -x '(objectclass=*)'
ldapsearch -x -D uid=fiona,dc=people,dc=example,dc=org \
           -w notverysecret '(objectclass=*)'
ldapsearch -x -D uid=fiona,dc=people,dc=example,dc=org \
           -w notverysecret -z 3 '(objectclass=*)'
ldapsearch -x -D uid=fiona,dc=people,dc=example,dc=org \
           -w notverysecret -z 30 '(objectclass=*)'
```

Note that hitting a size limit is not a fatal error. Results are still returned up to the limit.

4. Add a limits clause to give members of the 'kiwi' group a soft size limit of 14. Test to make sure that each class of user now gets the correct number of results.

There are several other forms of limit that can be used to protect large servers.

As with ACLs, it is generally best to apply limits to *groups* rather than to individual user DNs. Users can then be given the appropriate limits by assigning them to the right group, which is much easier than making routine changes to the server configuration.

Be careful when applying limits: any ID used by a replication consumer must be able to search without limit. Some applications (including some Linux getXbyY() calls) can also be broken by limits.