Access Control Lists (ACLs)

Broadly, an ACL associates

- What is being operated on?
- Who is requesting the operation?
- What type of operation is requested?

See slapd.access man page for details

"Access to <What> by <Whom> <Privilege>"

What

```
dn[.dnstyle]=<dnpattern>
filter=<ldapfilter>
attrs=<attrlist> [val[/matchingRule][.attrstyle]=<attrval>]
= all entries
<dnstyle> = (base|one|sub|children|regex)
<attrlist>=(<attr>|[(!|@)]<objectClass>)[,<attrlist>]
<attrstyle>=(base|one|sub|children|regex)
```

Who

- * - everyone
- anonymous = unauthenticated clients
- users = all authenticated clients
- self = users whose DN matches the target entry
- dn = explicit DN
- dnattr = an DN-valued attribute in the entry
- group = the list of members in a group entry
- peername = the socket address of the client
- sockname = the socket address of the server listener
- domain = the DNS name of the client
- sockurl = the URL of the server listener
- set = ACL sets
- ssf = overall security strength factor
- transport_ssf = transport level SSF
- tls_ssf = tls-specific SSF
- sasl_ssf = SASL-specific SSF

SSF - Security Strength Factor

The server uses Security Strength Factors (SSF) to indicate the relative strength of protection. A SSF of zero (0) indicates no protections are in place. A SSF of one (1) indicates integrity protection are in place. A SSF greater than one (>1) roughly correlates to the effective encryption key length. For example:

DES = 56
3DES = 112
AES = 128, 192, or 256

A number of administrative controls rely on SSFs associated with TLS and SASL protection in place on an LDAP session. Security controls disallow operations when appropriate protections are not in place. For example:

```
security ssf=1 update_ssf=112
```

This requires integrity protection for all operations and encryption protection, 3DES equivalent, for update operations (e.g. add, delete, modify, etc.).

Privilege Access Level

Characters in parenthesis are accepted abbreviations and can be used in place of their respective full-word identifiers.

- (0) none = no access
- (d) disclose = information disclosure on error (default)
- (a) auth = authentication
- (c) compare = Compare operations
- (s) search = search filter evaluation
- (r) read = search results
- (w) write = modifications
- (m) manage = all access including administrative access

Each level includes the preceding ones

Modify Privilege Rights

- Use (+ | - | =) modifiers to add, remove, or replace rights.
  Note: +0 is standalone and cannot be combined with any other access privileges.

Evaluation

ACLs are evaluated in the order in which they appear in the configuration

- Most specific rules must appear before general rules
- Evaluation usually stops at the first match

Additional controls may be specified to alter the evaluation sequence

Evaluation controls

- Stop = stops evaluation at the current rule
- Continue = keep examining other <who> clauses within the same <what> rule
- Break = keep examining other <what> rules

Example for slapd.conf

```
access to dn.subtree="dc=example,dc=com"
  by * =cs break
access to dn.subtree="ou=people,dc=example,dc=com"
  by * +r
```

- Gives Compare and Search access to all entries in the tree to all users
- Adds Read access for all entries in the “ou=people” subtree

Example for slapd.d

```
olcAccess: to dn.subtree="dc=example,dc=com" by * =cs break
olcAccess: to dn.subtree="ou=people,dc=example,dc=com" by * +r
```

Service-Type ACLs

Access rules for replication or service-type accounts. Since the rule is very specific and in the case of the replication user, will be evaluated often, this rule should be near the top of the ACL.

Common Directory Structure/Entries

```
ou=admin,dc=example,dc=com
  + cn=replicator
  + cn=updater
ou=groups,dc=example,dc=com
  + cn=read_all
      member: cn=replicator,ou=admin,dc=example,dc=com
  + cn=write_all
      member: cn=updater,ou=admin,dc=example,dc=com
```

Replication/Service Account ACL

```
access to *
```

`by group="cn=read_all,ou=groups,dc=example,dc=com" read stop`	Read access to everything by members of the cn=read_all group
`by group="cn=write_all,ou=groups,dc=example,dc=com" write stop`	Write access to everything by members of the cn=write_all group
`by * break`	What doesn't match will be evaluated by the next rule

Examples (slapd.conf vs. slapd.d)

Note: slapd.d examples may word wrap

SLAPD.CONF (default)

```
access to dn="" by * read
access to *
  by self write
  by users read
  by anonymous auth
  by sockurl="^ldapi:///$" write
```

SLAPD.D (default)

```
olcAccess: {0}to dn.base="" by * read
olcAccess: {1}to * by self write by users read by anonymous auth by
sockurl.exact="^ldapi:///$" write
```

- Allow all to read rootDSE
- Entries can write to self
- Authenticated users may read
- Unauthenticated users may use attributes for authentication
- Unix domain sockets may write

SLAPD.conf

```
Access to attrs=userPassword
      by self write
      by anonymous auth
      by dn="cn=admin,dc=example,dc=com" write
      by * none
Access to dn.base=""
      by dn="cn=admin,dc=example,dc=com" read
Access to *
      by dn="cn=admin,dc=example,dc=com” write
Access to dn.subtree="dc=qa,dc=example,dc=com"
      by dn="cn=admin,dc=qa,dc=example,dc=com" read
      by * break
Access to dn.children="dc=qa,dc=example,dc=com"
      by self write
      by dn="cn=admin,dc=qa,dc=example,dc=com" write
      by dn.children="dc=qa,dc=example,dc=com" read
```

SLAPD.D

```
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
manage by * break
olcAccess: {1}to attrs=userPassword by self write by anonymous auth by
dn="cn=admin,dc=example,dc=com" write by * none
olcAccess: {2}to dn.base="" by dn="cn=admin,dc=example,dc=com" read
olcAccess: {3}to * by dn="cn=admin,dc=example,dc=com" write
```

```
olcAccess:          {4}to          dn.subtree="dc=qa,dc=example,dc=com"          by
dn="cn=admin,dc=qa,dc=example,dc=com" read by * break
olcAccess:   {5}to   dn.children="dc=qa,dc=example,dc=com"   by   self   write   by
dn="cn=admin,dc=qa,dc=example,dc=com" write by dn.children="dc=qa,dc=example,dc=com"
read
```

- Every authenticated user can change their own password
- The user cn=admin,dc=example,dc=com can read the base DN
- The user cn= admin,dc=example,dc=com can modify everything everywhere
- The user cn=admin,dc=qa,dc=example,dc=com can read the subtree dc=qa DN
- All dc=qa members can modify themselves
- The user cn=admin,dc=qa,dc=example,dc=com can modify all subtree dc=qa content
- All dc=qa members can read the subtree dc=qa content

ACL Creation Suggestion

Print a copy of the DIT from Apache Directory Studio. Use multiple colored pens to identify who you want to have what kind of access to what portion of the DIT. Note any areas that overlap as additional ACLs will be required to clearly define those areas.