Access Control Policies for LDAP

Andrew Findlay
Skills 1st Ltd

andrew.findlay@skills-1st.co.uk

January 2009

Why Access Control?

- Keep the bad guys out!
- Let the good guys in
  - Who can read
  - Who can modify

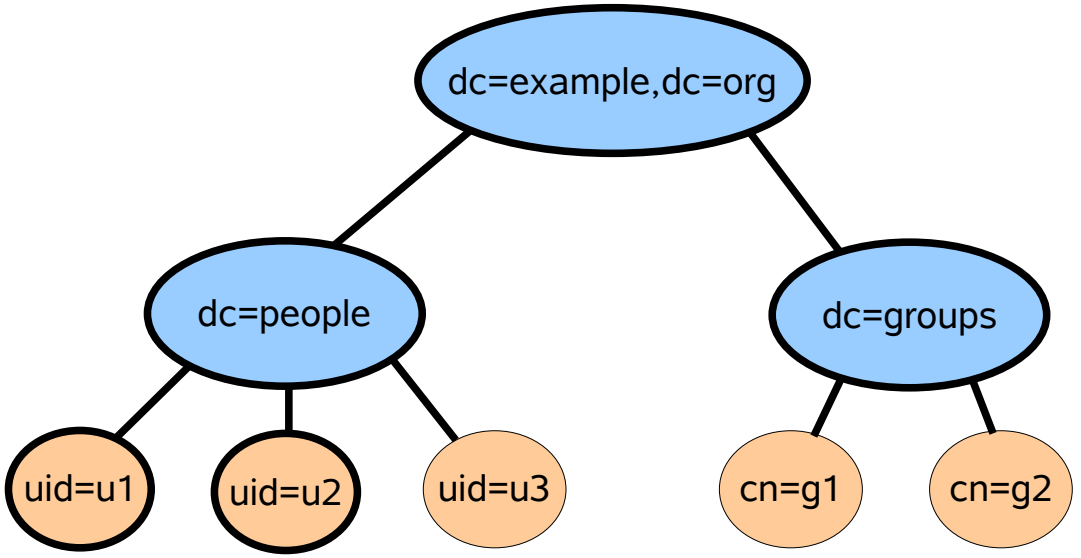
Concepts

- Subject – *who*
- Object – *what*
- Access – *permitted actions*
  - Read / Add / Delete entries
  - Read / Search / Modify attributes
- ACI – Access Control Item
- ACL – Access Control List

Design Process

- Define the requirements
  - Subjects: define groups
  - Objects: define categories
  - Allow for data management
  - Verify application requirements
  - Refine with examples
- Build a test suite
- Write ACLs

Simple Policies



- Read only
- Data admin
- Admin group

Design Principles

- ACLs are *programs*
- Have few ACLs
- Avoid routine ops involving ACLs
- Use attributes to trigger ACIs
- Write the tests *first*
- Don't mix grants and denys
- Give access to groups, not individuals

More Principles

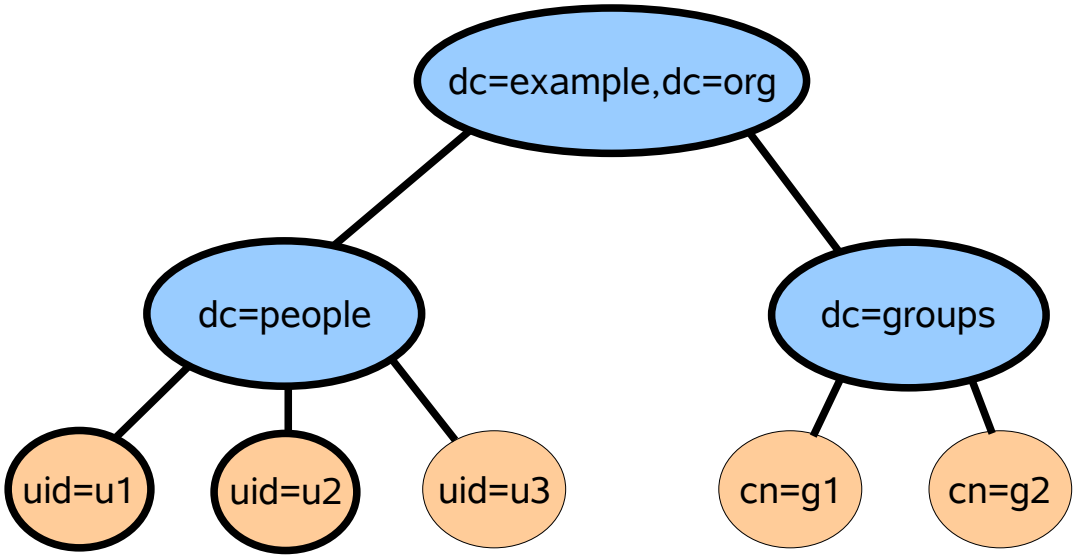
- Use DIT Content Rules
- Make DNs opaque
  - uniqeIdentifier=A674EC43
- Avoid spaces and punctuation in RDNs

Server capabilities

- IBM TDS
  - ACLs in DIT, at the control point
  - Filters
- Sun / Netscape / Red Hat / Fedora
  - ACLs in DIT, anywhere above control point
  - Filters, macros
- OpenLDAP
  - ACLs outside DIT, program-like
  - Filters, regular expressions, sets...

Example: user registry

- To authenticate users: ID and password
- Public read
- Users can change their own passwords
- Passwords not readable by anyone



ACLs for TDS

```
dn: dc=example,dc=org
changetype: modify
add: ibm-filterAclEntry
ibm-filterAclEntry: group:CN=ANYBODY:
 (objectclass=*):normal:rsc
ibm-filterAclEntry: access-id:cn=this:
 (objectclass=*):at.userPassword:grant:w
```

ACLs for Sun / Netscape

```
dn: dc=example,dc=org
changetype: modify
add: aci
aci: (targetattr != "userPassword")
  (version 3.0; acl "Make public objects visible";
     allow (read, compare, search)
     (userdn = "ldap:///anyone") ;)
aci: (targetattr = "userPassword")
  ( version 3.0; acl "Users change own passwords";
     allow (write)
     (userdn = "ldap:///self") ;)
```

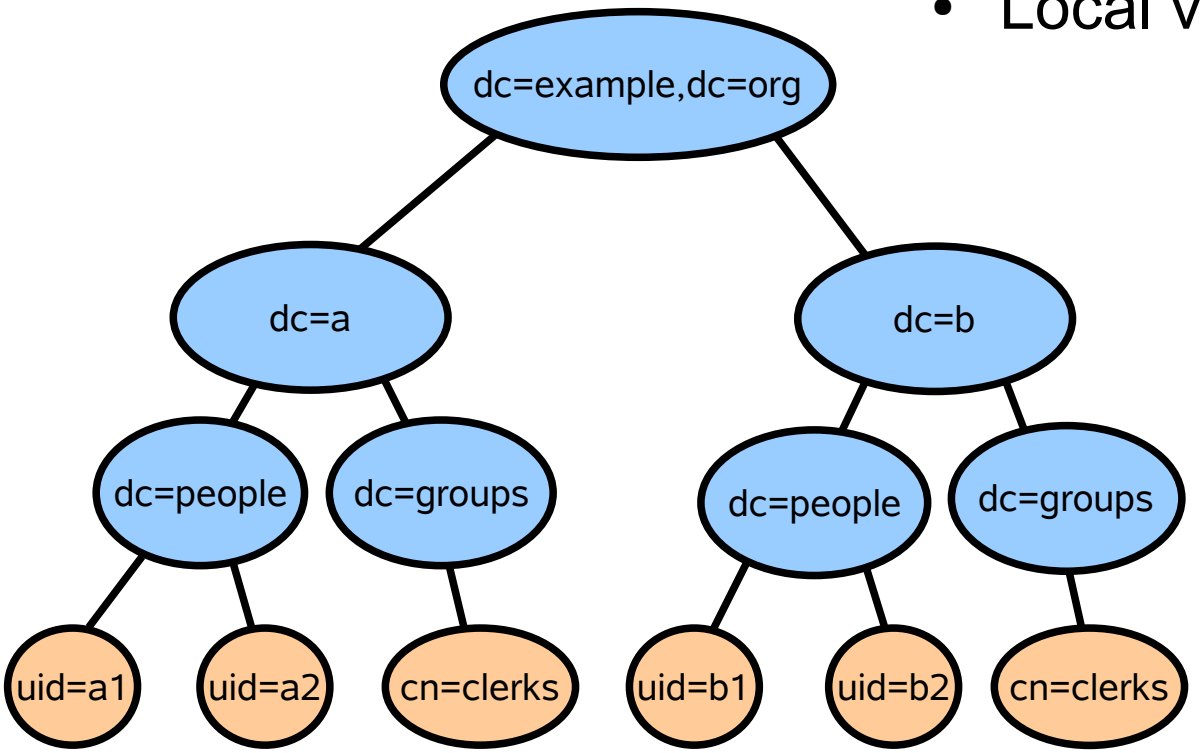
ACLs for OpenLDAP

```
access to attrs="userPassword"
        by self =w
        by * auth

access to *
        by * read
```

Example: Local Visibility

- Visibility attribute
- Local visibility by default



ACLs for TDS

- Needs an ACL in each department entry
- Identify local users with a dynamic group

```
dn: cn=users,dc=groups,dc=a,dc=example,dc=org
changetype: add
objectclass: groupOfURLs
objectclass: ibm-dynamicGroup
cn: users
memberURL: ldap:///dc=people,dc=a,dc=example,dc=org??
  sub?(objectclass=*)

dn: dc=a,dc=example,dc=org
changetype: modify
replace: ibm-filterAclEntry
ibm-filterAclEntry:
  group:cn=users,dc=groups,dc=a,dc=example,dc=org:
  (objectclass=*):normal:rsc
```

ACLs for TDS

- Global ACL: passwords and public entries

```
dn: dc=example,dc=org
ibm-filterAclEntry: access-id:cn=this:
 (objectclass=*):at.userPassword:grant:w
ibm-filterAclEntry: group:CN=ANYBODY:
 (exampleVisibility=public):normal:rsc
```

ACLs for Sun / Netscape

- Macro selects same-department users

```
dn: dc=example,dc=org
aci: (target="ldap:///($dn),dc=example,dc=org")
  (targetattr != "userPassword")
  (version 3.0; acl
    "Users see entries in their own department";
    allow (read, compare, search)
    (userdn =
"ldap:///dc=people,[$dn],dc=example,dc=org??sub?")
  ;)
```

ACLs for Sun / Netscape

- Filter selects public entries

```
dn: dc=example,dc=org
aci: (targetfilter =
          "(exampleVisibility=public)")
  (targetattr != "userPassword")
  (version 3.0;
   acl "Make public objects visible to all";
   allow (read, compare, search)
  (userdn = "ldap:///anyone")
 ;)
```

ACLs for OpenLDAP

```
access to dn.subtree="dc=example,dc=org"
       attrs="userPassword"
       by self =w
       by * auth

access to filter="(exampleVisibility=public)"
       by * read

access to dn.regex="(dc=[^,]+,dc=example,dc=org)$"
       by dn.subtree,expand="dc=people,$1" read
       by * break

access to * by * none
```

Controlling DIT Content

- For delegated administration
- ACLs should only allow write for correct object type
  - OpenLDAP, Netscape OK. TDS fails.
- Need to control auxiliary classes: DIT Content Rule

```
ditcontentrule ( 2.16.840.1.113730.3.2.2
        NAME 'dcrPerson'
        DESC 'Control inetOrgPerson entries'
        AUX strongAuthenticationUser
        )
```

Attribute sets for OpenLDAP

- Use object class to define set
- Remember to give access to “entry”

```
objectclass ( 1.2.826.0.1.3458854.666.3.1
   NAME 'attrsetAnonVisible'
   DESC 'Attributes visible to anon users'
   AUXILIARY
   MAY ( objectclass $ cn $ sn $ displayname $
         mail $ uniqueIdentifier )
   )

access to filter="(objectclass=person)"
        attrs="entry,@attrsetAnonVisible"
        by * +rsc break
```

Gotchas

- Hard to hide entries entirely
  - Detection by error message
  - OpenLDAP can protect leaf entries
  - Others have no protection
- Hard to control content of new entries
  - OpenLDAP can do it
  - Sun / Netscape has some control
  - TDS has none

Summary

- Access control needs care
- Difficulty can rise fast with policy size
- Test-driven development
- Design patterns
- Read the paper

Andrew Findlay

Andrew.Findlay@skills-1st.co.uk