Directory-enabling Your Application

University of Hawaii
Russell Tokuyama 02/20/02

University of Hawaii © 2002
Dimensions of LDAP

- Information model
- Naming model
- Functional model
- Security model
Information Model

- Schema - structured
- Entry - record, has attributes
- Attribute - field, data about the entry
- Type - syntax, kind of data stored
Information Model (cont.)

- **Schema**
  - What data can be stored
    - Object classes – groups of attributes
  - Attributes in an entry
    - Name
    - Type
    - Value
Information Model (cont.)

- Entry
  - Attributes
  - Types
  - Values
Naming Model

- How things are identified
- DN - distinguished name
  - unique name for an entry
  - uhUuuid=1234, ou=People, dc=hawaii, dc=edu
- RDN - relative distinguished name
  - composed of one or more attributes of the entry
  - uhUuuid=1234
Namespace

- What do you call things
- How things are grouped
- Hierarchical DIT
  - Contains entries
  - Entries contain entries

\[
dc=\text{hawaii}, dc=\text{edu}
\]

\[
ou=\text{people}
\]

\[
ou=\text{uh}
\]

\[
ou=\text{lcc}
\]

\[
ou=\text{uhh}
\]
Functional Model

- How information accessed
- Searching - area, filter, and scope, types of comparisons
- Update - add, delete, modify, modify RDN
- Connecting – bind, unbind, abandon
Searching

- Search Attribute(s)
  - Exact match
  - Approximate match
    - partial key
    - regular expression
  - Substring
Searching (cont.)

- **Search Attribute(s)**
  - Exact match

(sn=Doe)

(& (sn=Doe) (givenName=John) )
Searching (cont.)

- Search Attribute(s)
  - Approximate match

(sn~ = road)

(& (sn~ = Doe) (givenName = John) )
Search Filters

- Search Attribute(s)
  - Substring

(sn=do*)

(& (sn=r*oa*) (givenName=sam*))
More examples

(& (sn=do*) (! (givenName=jo*)))

(& (givenName=Jo*n) (|(sn=Do*)(sn=Sm*)))))
Security Model

- Access control
- Who gets to see which attributes
- Who gets to change things
- Roles
- Views
Access Control

- FERPA (Federal Education Rights and Privacy Act)
- Students not visible anonymously
  - White pages
  - User authentication (requires special DN)
LDAP C API – RFC 1823

- `ldap_bind()` - authenticate to directory server
- `ldap_result()` - get result from previous operation
- `ldap_search()` - search for directory entries
LDAP C API (cont.)

- `ldap_compare()` - does entry contain attribute value?
- `ldap_modify()` - change entry
- `ldap_unbind()` - end session
LDAP C API (cont.)

- `ldap_add()` - new entry
- `ldap_delete()` - remove entry
- `ldap_modrn()` - rename entry
ldap_search - Lookups

- Host:port of LDAP server
- Search base (where to look)
- Scope (how much of tree)
- Search filter (what to look for)
- Attributes to return
- Attributes only
Using the LDAP C API

- Create session and open connection
- Set parameters, call LDAP function, and get results
- Process results
- End session
#include <ldap.h>

/* Create session and open connection */
ld = ldap_init(ldap_server, ldap_port);
rc = ldap_simple_bind_s(ld, BIND_DN, BIND_PW);

/* Set params, call LDAP function, & get results */
rc = ldap_search_s(ld, ldap_search_base,
   LDAP_SCOPE_SUBTREE,
   ldap_filter, NULL, 0, &results);
/* Process results */
entry = ldap_first_entry(ld, results);
for ( attr = ldap_first_attribute(ld, entry, &element);
    attr != NULL;
    attr = ldap_next_attribute(ld, entry, element) ) {
    if ((vals = ldap_get_values(ld, entry, attr)) != NULL)
        for (i = 0; values[i] != NULL; i++)
            printf("  %s : %s \n", attr, 

Another way to search

- LDAP URL (RFC 2255)


/ou=people,dc=hawaii,dc=edu??one?(sn=Doe)
What will I get back?

- Zero or more entries
- Attributes
- Value(s) for each attribute present
Anything but C

- Java / JNDI
- Perl
- Visual BASIC
- PHP
- Cold Fusion
- Python
import javax.naming.*;
import javax.naming.directory.*;

// Create session and open connection
env.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, ldapUrl);
env.put(Context.SECURITY_AUTHENTICATION,
        "simple");
conn = new InitialDirContext(env);
// Set params, call LDAP function, & get results
searchCtls.setReturningAttributes(attributes);
searchCtls setSearchScope(
    SearchControls.SUBTREE_SCOPE);
results =
    conn.search(ldapBaseDN, searchFilter.toString(),
    searchCtls),
// Process results
Attributes attrs = entry.getAttributes();
for (int i=0; i < attributes.length; i++) {
    Attribute attrib = attrs.get(attributes[i]);
    for (vals = attrib.getAll(); vals.hasMoreElements(); )
        value.append((String) val.nextElement());
    System.out.println(attributes[i] + " = " +
                        value.toString());
}
Results Returned:
  uid = russ
  givenName = Russell
  sn = Tokuyama
  cn = Russell C Tokuyama
  eduPersonOrgDn = uhm
  eduPersonAffiliation = staff, student
  mail = russ@hawaii.edu
  uhPreferredMail = russ@hawaii.edu
# Perl Sample

# Sample by Bin Zhang of KCC Library
# Perl LDAP interface
use Net::LDAP;
use Net::LDAP::Util qw/ldap_error_text/;

# Create session and open connection
my $ldap = Net::LDAP -> new($server);
# Set params, call LDAP function, & get results
my $result = $ldap->search(base => 'ou=people, dc=hawaii,dc=edu', filter => "uid=$uid", );

# Process results
my $entry = $result->entry(0);
foreach my $attr ($entry->attributes) {
    foreach my $value($entry->get_value($attr)) {
        print("$attr\t= $value\n");
    }
}
Results returned:

edupersonorgdn  = uhm
uid            = russ
mail           = russ@hawaii.edu
uhpreferredmail = russ@hawaii.edu
cn             = Russell C Tokuyama
sn             = Tokuyama
givenname      = Russell
edupersonaffiliation  = staff
edupersonaffiliation  = student
‘ Set reference to ActiveX Data Objects 2.5

‘ Create session and open connection
Dim con As New Connection
Dim Idap As New Command
con.Provider = “ADsDSOObject”
con.Open “Active Directory Provider”, “”, “”
Set Idap.ActiveConnection = con
'Set params, call LDAP function, & get results
baseDN = "<LDAP://" & ldapHost & "/" & searchBase & ">
ldap.CommandText = baseDN & ";" & _
searchFilter & ";" & _
returnAttributes & ";" & _
searchScope
Set rs = ldap.Execute
<%  
<!-- Create session and open connection -->
$ds = ldap_connect("ldap1.its.hawaii.edu");
$rs = ldap_bind($ds);

<!-- Set params, call LDAP function, get results -->
$dn = "ou=People, dc=hawaii, dc=edu";
$filter = "(uid=$username)"
<!-- Set params, call LDAP function, get results -->
$justthese = array( "ou", "sn", "givenname",
    "eduPersonAffiliation", "title",
    "telephoneNumber",
    "facsimiletelephoneNumber",
    "physicalDeliveryOfficeName");
$sr=ldap_search($ds, $dn, $filter, $justthese);
$info = ldap_get_entries($ds,$sr);

$info = ldap_get_entries($ds,$sr);

echo "<table border="0" >
<tr><td><b>Name:</b></td><td>". $info[0]["cn"] [0] . "</td></tr>
<tr><td><b>Affiliation:</b></td><td>"." . $info[0]["eduPersonAffiliation"] [0] . "</td></tr>
<tr><td><b>Email:</b></td><td>"." . $info[0]["mail"] [0] . "</td></tr>";
<cfldap action="QUERY"
   name="result"
   server="ldap1.its.hawaii.edu"
   port="389"
   start="ou=people, dc=hawaii, dc=edu"
   attributes="uid, eduPersonAffiliation, uhSSN, eduPersonOrgDn, password, dn"
   filter="(uid=#trim(form.username)#)">
<cfset #accountType# = GetToken(#result2.eduPersonAffiliation#, 1, ",")>
# Create session and open connection
conn = ldap.open(ldapHost, ldapPort)

# Set params, call LDAP function, & get results
filter = '(uid=' + itsUsername + ')
rc = conn.search(searchBase, searchScope, filter, attrList)
results = conn.result(rc)
resultData = results[1]
# Process results

```python
attrs = resultAttrs.keys()
for i in range(len(attrs)):
    print attrs[i], ‘:’
    attrValues = resultAttrs.get(attrs[i])
    for j in range(len(attrValues)):
        print '    = %s' % attrValues[j]
```
Authenticating users

- Bind and lookup the uid
- Get the user’s DN
- Rebind as the user (DN and pw)
  - Blank pw = anonymous (caution)
- Search for anything in user’s entry
- Don’t need to lookup user’s pw
Programming hints

- KISS - Focus on just what you need
- Refactor for modularity
- Choose meaningful names
  - Use server Not svr (anybody’s guess)
- Test, test, test
- Re-read the documentation
- How do you know what went wrong?
Resources

- IETF RFC
  - 1823 – The LDAP Application Program Interface
  - 2255 - The LDAP URL Format
Resources (cont.)

- Implementing LDAP by Mark Wilcoxon, 1999, Wrox Press
Resources (cont.)

Resources (cont.)

- Java
- Perl - http://cpan.perl.org
- VB
Resources (cont.)

- Python - http://www.python.org/
- ITS Brown Bags
  - http://www.hawaii.edu/brownbags/ldap
- UH LDAP Directory Service
  - http://www.hawaii.edu/ldap/
Feedback

- ITS Systems Services
  - Russ Tokuyama <russ@hawaii.edu>