Internet infrastructure

Prof. dr. ir. André Mariën
Lightweight Directory Access Protocol
LDAP

Introduction
Directory

- Directory: searchable information repository
- The best known non-IT example is a telephone directory (the book)
  - Contains names, addresses, telephone numbers
  - Searchable by location and name
- Electronic variants
  - Also search by number, subparts, ZIP codes, ...

IT directories:
- Network configuration
  - Host to IP translation, user/password information, ...
- User information:
  - Telephone, email, certificates, physical location
- Security:
  - Authentication, authorization, SSO, PKI
- Resources:
  - Calendar, room reservations
Increasing requirements

• Consolidation of directories (N+1 problem)
  – Each new product comes with an extra directory
• Centralized management
  – Master data idea: manage once
• Consistency of information
  – Keeping copies in sync
• Access time
  – Critical resource: in stream
Example scenario

Web server

Web server

Application server

Personalization

Network logon

Certificate (SSLv3)

Authorization

Personalization

LDAP
Data modeling

- Data is modeled as a tree
- Multiple trees are possible (forest)
- Information is stored in ALL the nodes
- Nodes have a direct address
  - The address is content based
  - As much as possible: get data with direct address
Data modeling: example

• Users live in a country, belong to an organizational entity, and have a name

• Data of that user can be found simply:
  – Get data of “country=be, organizationunit=sales, name=V.Erkoper”

• The organizational unit can have data on its own, which can be obtained via
  – Get data of “country=be, organizationalunit=sales”

• The example assumes a tree with structure:
  – Top node
  – Country nodes
  – Organization units
  – Persons

• Any node’s properties can be accessed directly

• Need to understand the tree structure
Directory Information Tree

DIT graph

- o=kuleuven.ac.be
- ou=cs
- l=200A
- cn=André Mariën

18/03/2014

(c) A. Mariën
Directory Information Tree

• DIT is made up of entries or objects
• The names are based on standards
  – o : organization
  – C : country
  – ou : organizational unit
  – l : location
  – cn: common name
• Each of these is associated with a specific type of data
LDAP Objects

- An LDAP directory is built from objects
- Objects have an object identifier (OID)
- Objects have unique names
- Objects have attributes

Examples:
- People
- Resources: rooms, cars
- Groups: mailing groups
- Roles: for authorization
- Systems: machines, laptops, PDAs
- Rights: for authorization
- Certificates
Objects and Attributes

• Objects have attributes
  – Key/value pairs associated with the object

• Attribute have types
  – Predefined or custom (example: SSN)
  – Definition: syntax

• Attribute value: one or more values of the attribute type

• Order of attributes & values
  – undefined and implementation-dependent
  – MUST NOT be relied upon.

Examples

• Cn=billy the kid
• O=mafia
• C=krim
• Ou=blackmail
• Member=john
  member=paul
  member=ringo
  member=george
DIT and entries

- DIT
- Entries/Objects
- Attributes
- Values
Object Naming: DN - RDN

• Relative Distinguished Name (RDN)
  – One or more attribute values from the object form it’s relative distinguished name (RDN), which MUST be unique among all its siblings

• Distinguished Name (DN)
  – The concatenation of the RDNs of the sequence of entries from a particular entry to an immediate subordinate of the root of the tree forms that entry's DN
  – DN is unique in the tree
RDN

RDN: A,B
RDN P: A=1, B=2

RDN: C
RDN Q: C=1
Q

RDN: F, G
RDN R: F=1, G=1
RDN S: F=1, G=2

P
A=1, B=2

C=1, D=2
C=2, E=2

F=1, G=1, H=3
F=1, G=2, I=2
DN

DN P: A=1, B=2

DN Q: C=1, A=1

DN R: F=1, G=1, C=1, A=1, B=2
DN S: F=1, G=2, C=1, A=1, B=2
Important Attributes

• cn
• commonName
• name of an object
  – for a person, it is typically the person's full name
  – a webserver: its DNS name
  – a web certificate: an email address
Important Attributes

• o
  – organizationName

• ou
  – organizationalUnitName

• Attributes: address: c
  – country
  – two-letter ISO 3166 country code
Important Attributes

• l
  – localityName
  – locality: city, county, geographic region

• st
  – stateOrProvinceName
  – a state or province

• street
  – streetAddress
Important Attributes

• **member**
  – used in grouping constructs
  – also used for authentication and authorization
  – each value a DN of a 'member' object

• **owner**
  – links objects to object owner DN
  – example: devices to responsible
  – could be used as 'father' link in groups
Attributes

• person
• givenName
• initials
• generationQualifier
• sn
  – surname: the family name of a person.
• userPassword
Attributes: structural

• distinguishedName
  – a base type from which attributes with DN syntax inherit
  – not used as the name of the object itself

• uniqueMember

• name
  – attribute supertype
  – string attribute types for naming are derived from this; will not occur in an entry
Lightweight Directory Access Protocol

LDAP Search
Search Request

• **Where?** baseObject DN
• **Scope?** baseObject / singleLevel / wholeSubtree
• **Conditions?** filter
• **What?** list of attributes / '*': all
Search Request Scope

node

Whole subtree

One level
Filter: setting conditions

- boolean: and / or / not
- equalityMatch
- substrings: initial / any / final
- greaterOrEqual, lessOrEqual
- present
- approxMatch
- extensibleMatch: new in v3
String Representation of LDAP Search Filters

• RFC 2254

• Why an extra RFC?
  – LDAP defines a network representation
  – also wanted: search filters in a human-readable form

• String Search Filter Definition
  – prefix notation: ( op operand1 operand2 ...)
Simple Filter

Filter: attribute \textit{filtertype} value

Filtertype:
  - Equal: "="
  - Approx: "\sim="
  - Greater: ">="
  - Less: "<="
  - Present: "=*"

Examples:
  - (cn=Mickey)
  - (objectClass=person)
Wildcard matching

Wildcard symbol: *

substring  = attr "=" [initial] any [final]
initial    = value
any        = "*" *(value "*")
final      = value

Examples:
  (cn=*andre*)
  (c=*germ*)
Filter: boolean operators

• Prefix notation
  – filter = "(" filtercomp ")"
  – filtercomp = and / or / not / item
  – and = "&" filterlist
  – or = "|" filterlist
  – not = "!" filter
  – filterlist = 1*filter
Example filters:

• (cn=Babs Jensen)
• (!!(cn=Tim Howes))
• (&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))))
• (o=univ*of*mich*)
LDAP URLs
The LDAP URL Format

• RFC 2255: The LDAP URL Format
• defines a format for a LDAP Uniform Resource Locator (URL)
• models a LDAP search operation
URL Definition

ldapurl=
   "ldap://" [hostport]
   ["/"] [dn ["?"] [attributes]
       ["?"] [scope]
           ["?"] [filter] ["?"] [extensions]]
)

URL: elements
• hostport:
   – host:389
• starting point (base object):
   – dn: distinguishedName
• which attributes:
   – attributes = attrdesc *("," attrdesc)
• Scope = "base" / "one" / "sub"
   – node, one level, subtree

(c) A. Mariën
URL elements

Filter
• filter: text format search string
• additional (extra):
  – extensions = extension *(""," extension)
• default: (objectClass=*)
• all URL symbols: url encoded (%xx)

Authentication: "Bindname" Extension
• type: "bindname"
• value: DN of the directory entry to authenticate as
• dn == NULL string: unauthenticated access
Example queries

• Example 1
  • base object:
    – "o=University of Leuven, c=BE"
  • filter:
    – "(objectclass=*)"
  • all attributes
  • URL:
    – ldap://ldap.itd.umich.edu/o=University%20of%20Leuven,c=BE

• Example 2
  • return only attribute:
    – postalAddress
  • URL:
    – ldap://ldap.itd.umich.edu/ o=University%20of%20Michigan,c=US?postalAddress
Example queries

Example 3:
• subtree search
• common name: "Babs Jensen"
• all attributes:
• URL:
  ldap://host.com:6666/o=University%20of%20Michigan,c=US
  ?sub?(cn=Babs%20Jensen)

Example 4
• all direct children of the c=GB
• attribute: objectClass
• URL:
  ldap://ldap.itd.umich.edu/c=GB?objectClass?one
Example queries

Example 5
• attribute: mail
• object: "o=Question?,c=US"
• uses the URL encoding for character '?'
• URL:
  ldap://ldap.question.com/o=Question%3f,c=US?mail

Example 6
• interaction between LDAP and URL quoting mechanisms
• LDAP filter syntax:
  (int=\00\00\00\04)
• \ character must be URL encoded: \ -> %5c
• URL: 
  ldap://ldap.netscape.com/ o=Babsco,c=US?%(int=%5c00%5c00%5c00%5c00%5c04)
Example 7

- bindname extension
- DN for authentication
- %2C -> "," %5C
- URL:

  ldap:///??sub??

  bindname=cn=Manager%2co=Foo
Simple testing

• There are free directory browsers available
• There are some public directories online
• Available @ 18/03/2014
  – Pkcsldap.tttc.de
  – X500.de
• !!! Behave !!!
• A test infrastructure will be set up @KULeuven
Object Classes
Object Classes

• Object must belong >= 1 object class
• Objectclasses determine the possible object attributes
  – mandatory: union of all mandatory
  – optional: union of all optional \ mandatory

The objectClass attribute
• Object must have an objectClass attribute
• Values: the object classes with which the entry complies
• Example:
  – ObjectClass=top
  – ObjectClass=person
objectClass attribute

Object instance of class “oc”

- objectClass=oc
- att1=v1
- att2=v2
- att3=v3
- att3=v4

Object class Object for “oc”

- DN=oc
- Must=att1
- Must=att2
- May=att3
Object Classes

• *abstract* class (examples: "top" or "alias")
  – similar to Java abstract class

• *structural* object class
  – normal classes

• *auxiliary* object class
  – similar to interfaces in Java
Attribute inheritance

• objectClass has a superClass
• “top” is the base objectClass
• Inheritance: if objectClass \( o \) has superClass \( s \) then
  – all attributes of objectClass and those implied by \( s \) are possible attributes of \( o \)
  – any mandatory attributes of \( o \) or implied by \( s \) must be present
Attribute Inheritance

objectClass=oc
att1=v1
att2=v2
att3=v3
att3=v4

DN=oc
Must=att1
SUP=sup
DN=sup
Must=att2
May=att3
Object Class: BNF

ObjectClassDescription = "(" numericoid
    [ "NAME" qdescrs ] [ "DESC" qdstring ]
    [ "OBsolete" ]
    [ "SUP" oids ]
    [ ( "ABSTRACT" / "STRUCTURAL" / "AUXILIARY" ) ]
    [ "MUST" oids ] [ "MAY" oids ]
")"
Object Classes

Class “top”
  •  ABSTRACT
  •  MUST objectClass

Class: “person”
  •  SUP top
  •  STRUCTURAL
  •  MUST ( sn $ cn ) MAY ( userPassword $ telephoneNumber $ seeAlso $ description )

Class: “organizationalPerson”
  •  SUP person
  •  STRUCTURAL
  •  MAY ( title $ registeredAddress $ destinationIndicator $ telephoneNumber $ internationalISDNNumber $ facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $ postalAddress $ ou $ st $ l $ ... )
Object Classes

Class: “organization”
- SUP top
- STRUCTURAL
- MUST o
- MAY (userPassword $ searchGuide $ seeAlso $ businessCategory $ telexNumber $ teletexTerminalIdentifier $ telephoneNumber $ street $ postOfficeBox $ postalCode $ postalAddress $ st $ I $ description $ ...)

Class: “organizationalUnit”
- SUP top
- STRUCTURAL
- MUST ou
- MAY (userPassword $ searchGuide $ seeAlso $ businessCategory $ telexNumber $ telephoneNumber $ street $ postOfficeBox $ postalCode $ postalAddress $ st $ I $ description $ ... )
Object Classes

Class: “groupOfNames”
- SUP top
- STRUCTURAL
- MUST ( member $ cn )
- MAY ( businessCategory $ seeAlso $ owner $ ou $ o $ description )

Class: groupOfUniqueNames
- SUP top
- STRUCTURAL
- MUST ( uniqueMember $ cn )
- MAY ( businessCategory $ seeAlso $ owner $ ou $ o $ description )