Directory Services
(in particular LDAP)

Karst Koymans, Jaap van Ginkel

Informatics Institute
University of Amsterdam
(version 1.9, 2012/10/05 13:37:02)

Friday, October 12, 2012

Table of Contents

Common Directory Services

- Flat files (from BSD)
- NIS (Network Information Service from Sun)
  - was YP (Yellow Pages)
  - extended to NIS+
- NetInfo (NEXTSTEP - Mac OS X v10.4)
- Active Directory (Microsoft)
- LDAP (Lightweight Directory Access Protocol)

LDAP History (1)

- X.500 standard (1988)
  - Developed by CCITT (ITU-T)
  - Uses DAP (Directory Access Protocol)
    - Between DUA (Directory User Agent)
    - and DSA (Directory System Agent)
  - Based on OSI software
  - Revised in 1993

LDAP History (2)

- LDAP (Lightweight DAP) as simple access to X.500
  - LDAP v1 (RFC 1487) in 1993
  - LDAP v2 (RFC 1777) in 1995
- LDAP as replacement for X.500
  - LDAP v3 (RFC 2251) in 1997
    - Obsoleted by RFC 451i (i=0,...,9) in 2006

LDAP versus X.500 (1)

- LDAP v1 and v2
  - Works directly over TCP/IP
  - Use ordinary strings instead of ASN.1/BER in many cases
  - Simplifies BER in other cases

LDAP versus X.500 (2)

- LDAP v3
  - Simplifications from v1 and v2
  - Defines referrals
  - Uses SASL for security
  - Uses Unicode for internationalisation

Properties of directories

- Optimized for reads
- Distributed model for information storage
- Extendable information
- Advanced search capabilities
- Replication capabilities

LDAP models (1)

- Information model
  - Defines structures and data types
  - Defines the Directory Information Base (DIB)
- Naming model
  - How entries are referenced
  - Defines (Relative) Distinguished Names

LDAP models (2)

- Functional model
  - Defines the protocol
  - Defines what operations can be performed
- Security model
  - Provides authentication
  - Provides authorization
  - Provides confidentiality

LDAP models (3)

- How do LDAP models compare to the DNS environment?
  - Information model
    - Resource records
  - Naming model
    - Owner names (domain names)
  - Functional model
    - Query
  - Security model
    - Authentication, no authorization or confidentiality

Directory Information Tree

- A Directory Information Tree (DIT) is a tree
  - where the nodes are called Directory Entries
    - which each contain a set of attributes
    - where every attribute has a type and a value
- Directory Schemas are used to specify the allowed entries and attribute types
- LDIF (LDAP Data Interchange Format) is used to define specific entries

Naming Directory Entries

- An RDN (Relative Distinguished Name)
  - consists of a subset of attributes
  - that uniquely identifies the entry among its siblings
  - most of the time being a singleton subset
  - comparable to a primary key in a relational database
- An DN (Distinguished Name)
  - is a sequence of RDNs, separated by ","s
  - making the entry unique on the LDAP server

Special Attributes (1)

- the "objectClass" attribute is always present
  - objectClass defines valid attribute types for the entry
  - a "classic selfreference"
    - objectClass is always in the list
  - this attribute can be multivalued

Special Attributes (2)

- the "dn" attribute is not a real attribute
  - but is often presented as such
  - contains the distinguished name of an entry
  - is useful inside an LDIF representation

X.500 names and DNS labels

- A DNS domain name like "os3.nl." corresponds to
  - a distinguished name "dc=os3,dc=nl"
    - where "dc" is the domainComponent attribute
    - of an entry of objectClass: domain
    - which represents the LDAP server's naming context

Object classes

- An object class
  - specifies a name for the class
    - and its OID (object identifier)
  - specifies mandatory attribute types
  - specifies optional attribute types
  - is part of a class hierarchy (inheritance)

Attribute types

- An attribute type
  - uniquely specifies the name of the attribute type
    - and its OID (object identifier)
  - specifies whether it is single-valued or multi-valued
  - specifies the attribute syntax and matching criteria, for instance
    - testing for equality, ordering, ...

Attribute syntax

- specifies the kind of data for values (datatype)
- can be primitive or complex
- sets parameters for ranges or sizes

Directory schema (1)

- A directory schema specifies
  - available object classes
  - with the attribute types
  - and the attribute syntax

Directory schema (2)

- A schema can be written in several formats
  - ASN.1 schema format
  - LDAPv3 schema format
  - slapd.conf schema format

LDIF

- LDAP Data Interchange Format
- standard text file format describing directory entries
- defined in RFC 2849

LDAP wire format

- LDAP sends messages based on ASN.1
  - Abstract Syntac Notation One
- and uses a subset of BER for wire encoding
  - Basic Encoding Rules

LDAP operations (1)

- Authentication and control
  - bind
    - establish authentication state
  - unbind
    - abandon operations and close connections
  - abandon
    - abort earlier operation (by ID)

LDAP operations (2)

- Updates
  - add
    - create a new node
  - delete
    - remove a complete node
  - modify
    - change attributes or values at a node
  - modify DN
    - rename/move (R)DN

LDAP operations (3)

- Search and retrieve
  - search
  - compare
    - specialized search
    - can show nonexistence of an attribute

LDAP security

- Several security mechanisms are defined
  - None (anonymous access)
  - Clear text passwords
  - Kerberos authentication
  - SASL authentication
  - LDAP over SSL/TLS (STARTTLS or ldaps)

Searching

- A search operation has eight (!) parameters
- Replaces a non-existent read operation
- A read is a search restricted to only one DN

Search parameters (1)

- Base DN
- Scope
  - base
  - onelevel
  - subtree
- Treatment of aliases

Search parameters (2)

- Size limit (number of entries to return)
- Time limit (maximum time spent searching)
- Include attribute types and values or only types
- Search filter
- List of attributes to be returned

Search filters

- Boolean combination of atomic search filters
- Boolean operators allowed
  - & (Boolean AND)
  - | (Boolean OR)
  - ! (Boolean NOT)
- "(&(givenName=Niels)(|(l=Amsterdam)(l=Utrecht)))"

Atomic search filters (1)

- Equality
  - "(sn=van der ham)" matches "van der Ham"
- Greater Than or Equal To
  - "(age>=18)" matches "21"
- Less Than or Equal To
  - "(age<=21)" matches "21"

Atomic search filters (2)

- Substring
  - "(sn=*ham)" matches "van der Ham"
- Approximate
  - "(sn~=van der Hem)" matches "van der Ham"
- Presence
  - "(sn=*)" matches any entry with a sn attribute

Aliases

- Directory entries of objectClass "alias"
- Mandatory attribute "aliasedObjectName"
  - which contains a reference to another DN
- Could be compared with a CNAME in DNS

Referrals

- Directory entries of objectClass "referral"
- Optional attribute "ref"
  - which contains an LDAP URI
- Another option is "chaining"
  - Compare to recursion and iteration in DNS

LDAP URIs

- ldap://
  - fqdn:port/
    - distinguished_name
- ldaps://
  - fqdn:port/
    - distinguished_name