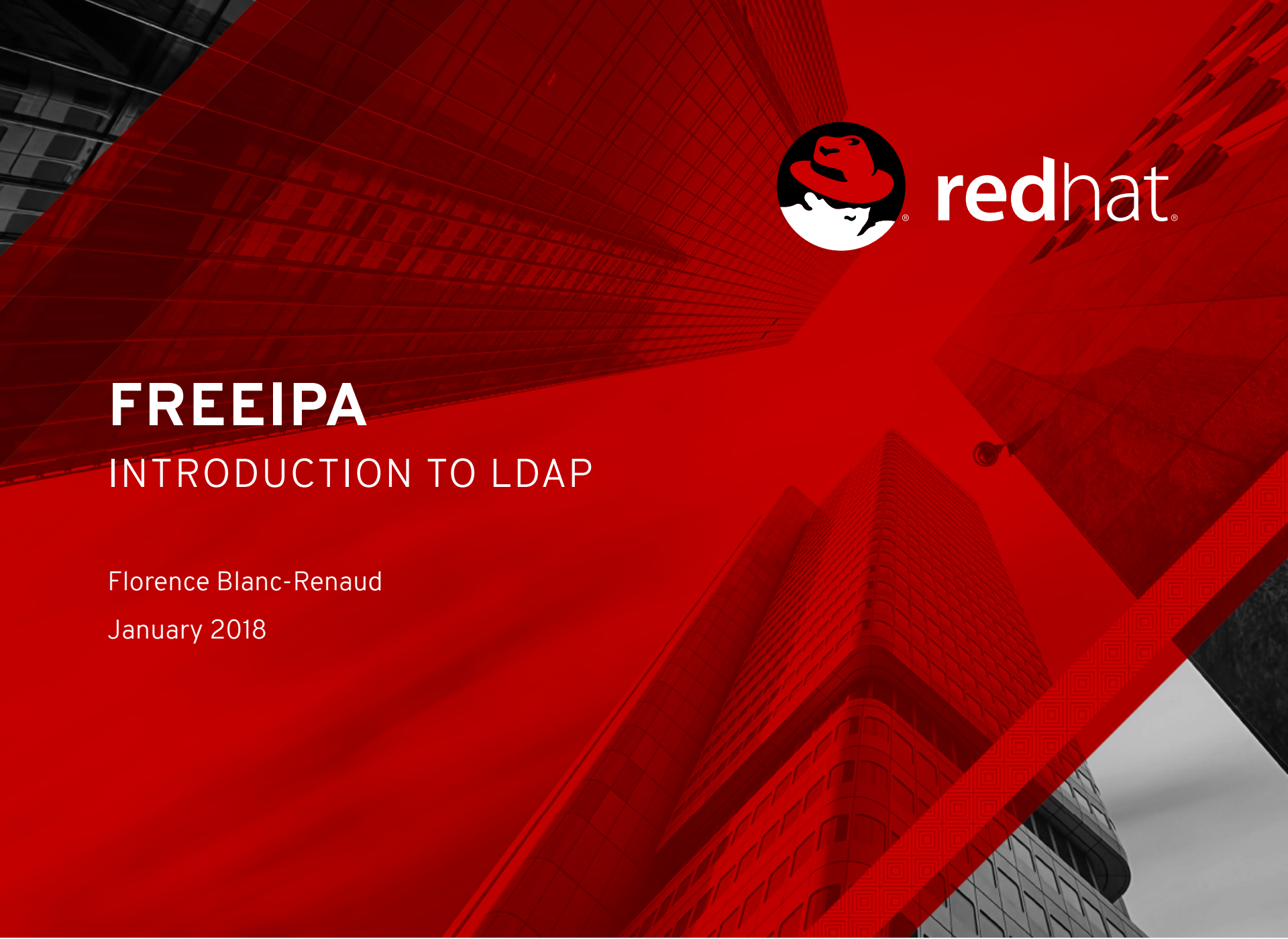


FREEIPA

INTRODUCTION TO LDAP

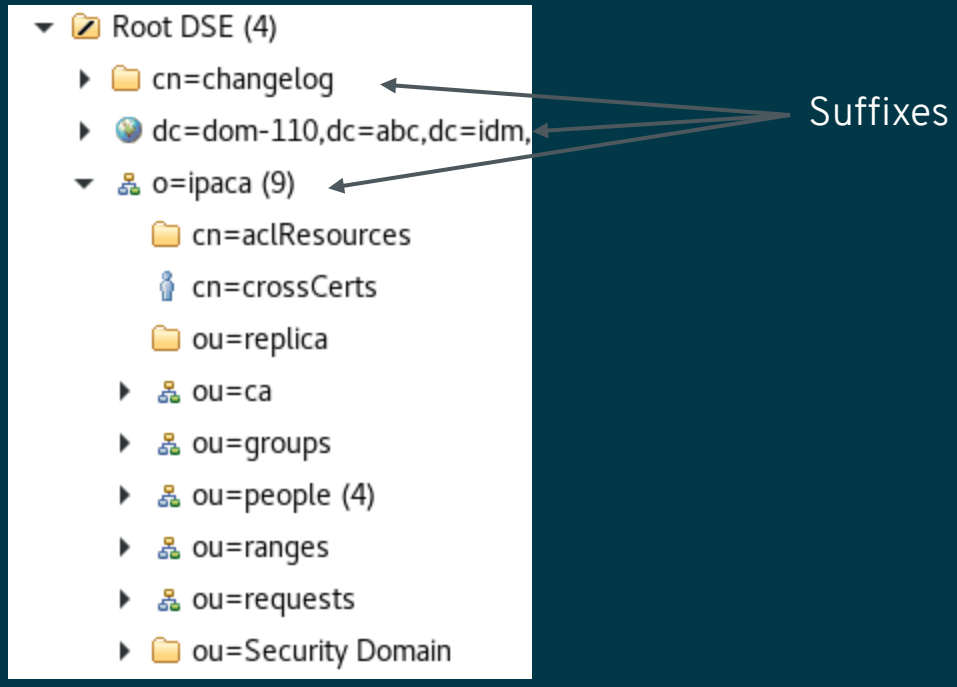
Florence Blanc-Renaud

January 2018

LDAP, AN APPLICATION PROTOCOL

LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

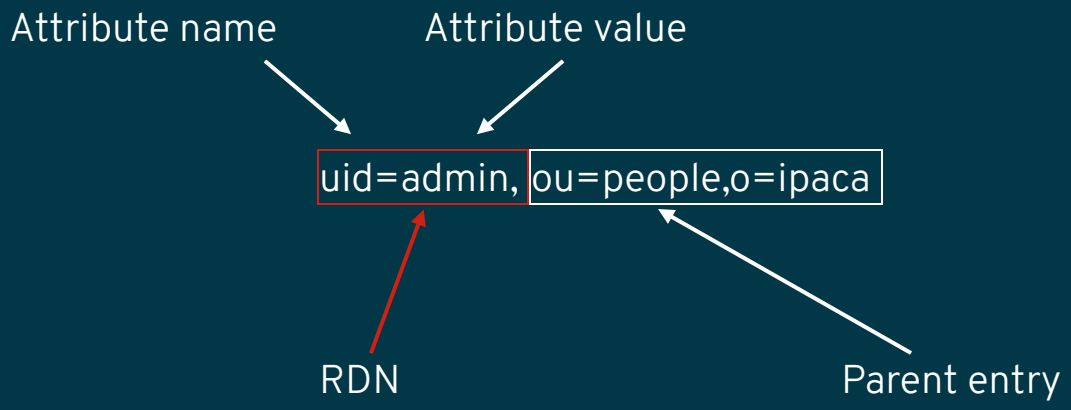
- Allows to access data stored in a server, from a client
- Data stored as entries in a tree
- Each entry is identified by its distinguished name (DN)
- Notion of hierarchy (parent entry / child entries)
  - top of the tree = the root DN (also called empty DN or null DN)
- Server contains one or more suffixes (or base DNs)



LDAP ENTRIES

NAMING

- DN is composed of RDNs (relative DNs)
- RDN contains one or more AVA attribute value assertion: attribute name = value
  see Anatomy of a DN for more information



LDAP ENTRIES

CONTENT

- Entry contains object classes and attributes
- An object class defines the mandatory and optional attributes
- An attribute can be single valued or multi valued
- An attribute contains data following a defined syntax (boolean, directory string, integer, generalized time...)
- The LDAP server schema defines the object classes and attributes

DN: uid=admin,ou=people,o=ipaca

Attribute Description	Value
objectClass	***cmsuser (structural)***
objectClass	***inetOrgPerson (structural)***
objectClass	***organizationalPerson (structural)***
objectClass	***person (structural)***
objectClass	***top (abstract)***
cn	**admin**
sn	**admin**
usertype	**adminType**
description	2;6;CN=Certificate Authority,O=DOM
mail	root@localhost
uid	admin
userCertificate	X.509v3: CN=ipa-ca-agent,O=DOM-
userPassword	SSHA-512 hashed password
userstate	1

LDAP SCHEMA (1/2)

Schema is accessible through LDAP protocol as cn=schema

Uses OIDs (object identifier) to uniquely identify schema definitions

ATTRIBUTE SYNTAXES

Define the kind of information that can be stored in an attribute

```
ldapSyntaxes: ( 1.3.6.1.4.1.1466.115.121.1.7 DESC 'Boolean' )
```

MATCHING RULES

How to make comparisons against attribute values

```
matchingRules: ( 2.5.13.27 NAME 'generalizedTimeMatch' DESC 'The rule evaluates to
TRUE if and only if the attribute value represents the same universal coordinated
time as the assertion value.' SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

LDAP SCHEMA (2/2)

ATTRIBUTE TYPES

Define the attributes (name, syntax, matching rules)

```
attributetypes: ( 2.5.4.3 NAME ( 'cn' 'commonName' )  SUP name EQUALITY
caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'RFC 4519' X-DEPRECATED 'commonName' )
```

OBJECT CLASSES

Define collections of attributes: MUST = mandatory, MAY = optional

An object class can inherit from another object class: SUP = superior object class

```
objectclasses: ( 2.16.840.1.113730.3.8.4.14 NAME 'ipaEntitlement' DESC 'IPA
Entitlement object' AUXILIARY MUST ipaEntitlementId MAY ( userPKCS12 $
userCertificate ) X-ORIGIN 'IPA v2' )
objectclasses: ( 2.16.840.1.113730.3.8.4.4 NAME 'ipaUserGroup' DESC 'IPA user group
object class' SUP nestedGroup STRUCTURAL X-ORIGIN 'IPA v2' )
```

LDAP OPERATIONS

- ABANDON: abort the processing of an operation
- ADD: create a new entry - parent entry must exist and entry must conform to the schema
- BIND: authenticate
- COMPARE: compare a value with the entry's attribute value
- DELETE: delete an existing leaf entry
- MODIFY: modify the content of an existing entry (add an attribute value, remove an attribute value or modify an attribute value)
- MODDN: move or rename an entry
- SEARCH: search for entries matching criterias or read an entry
- UNBIND: reverse operation of BIND

LDAP PROTOCOL

AN EXTENSIBLE PROTOCOL

- Extended operation: a generic op allowing to define new operations not described in the original specification (for instance StartTLS, Cancel or Password modify)
- Controls: appended to requests or responses, allow to modify the behavior (for instance Subtree delete control, Sort request control...)

CLIENT TOOLS

COMMAND-LINE INTERFACE

- ldapsearch
- ldapadd
- ldapmodify
- ldapdelete
- ldapcompare

- common options:
  - -h host
  - -p port
  - -H URI

    `ldap://host:port`
  - -D bind DN
  - -w password

GRAPHICAL TOOLS

- Apache Directory Studio, JXplorer, ...

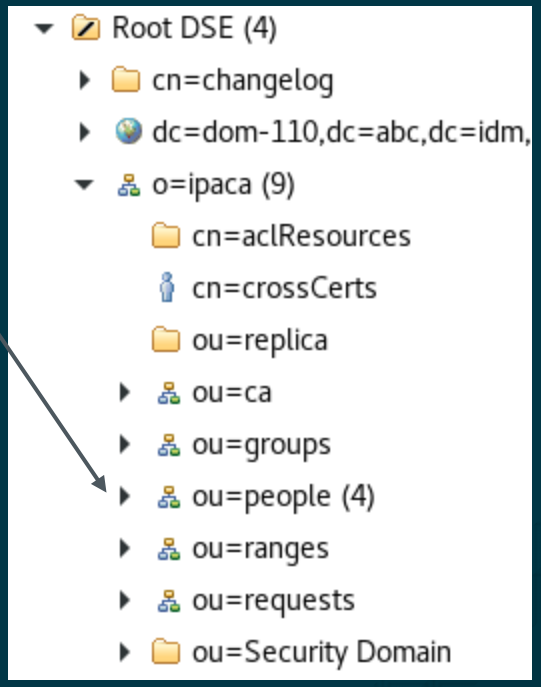
CLIENT TOOLS

AUTHENTICATION

- anonymous
- simple bind: username + password (-D / -w)
- SASL mechanism:
  - external: authentication with a user certificate. Need to match the content of the certificate with a user (-Y EXTERNAL, define $LDAPTLS_CACERTDIR, $LDAPTLS_KEY and $LDAPTLS_CERT)
  - GSSAPI: authentication with a Kerberos Ticket. Need to match the principal name with a user (-Y GSSAPI)
  - other SASL mechanisms exist but are less frequent (anonymous, CRAM-MD5, DIGEST-MD5...)

LDAPSEARCH

- ldapsearch -b ou=people,o=ipaca -s sub "(uid=admin)" dn uid
  - search base: look for entries below this base DN
  - scope: base | one | sub | children
  - search filter: return only entries matching the search criteria
  - requested attributes



LDAPMODIFY

- accepts either a LDIF file containing modifications or reads operations from standard input
- LDIF: LDAP data interchange format - RFC 2849
- Example:

```
ldapmodify -D "cn=directory manager" -w Password -H ldap://host:port
dn: uid=jdoe,cn=users,cn=accounts,dc=example,dc=com
changetype: modify
add: description
description: This is the user entry for John Doe
```

- changetype: add, delete, modify, modrdn, moddn

389-DS ACCESS LOG

- Stored in /var/log/dirsrv/slapd-DOMAIN/access
- Displays the connection established and the operations performed

```
[03/Jan/2018:09:27:57.702570690 +0100] conn=5314 fd=126 slot=126 connection from ::1
to ::1
[03/Jan/2018:09:27:57.703414600 +0100] conn=5314 op=0 BIND dn="cn=directory manager"
method=128 version=3
[03/Jan/2018:09:27:57.703768482 +0100] conn=5314 op=0 RESULT err=0 tag=97 nentries=0
etime=0 dn="cn=directory manager"
[03/Jan/2018:09:27:57.704093294 +0100] conn=5314 op=1 SRCH base="" scope=0 filter="
(objectClass=*)" attrs="namingContexts"
[03/Jan/2018:09:27:57.706176299 +0100] conn=5314 op=1 RESULT err=0 tag=101
nentries=1 etime=0
[03/Jan/2018:09:27:57.706591053 +0100] conn=5314 op=2 UNBIND
[03/Jan/2018:09:27:57.706685750 +0100] conn=5314 op=2 fd=126 closed - U1
```

corresponds to

```
ldapsearch -h localhost -p 389 -D cn=directory\ manager -w Secret123 -b "" -s base
namingcontexts
```

ACCESS CONTROL

- Implementation is specific to each Directory server
- Before an operation is performed, the server checks if the user is allowed to access and modify the data
- ACI (access control information) can be defined in the entry as an aci attribute or in one of the parent entries
- Syntax: aci: (target)(version 3.0;acl « name » ;permissionbind_rules;)

- Example: in the entry $SUFFIX
  - A user is allowed to write the attribute usercertificate in its own entry:
  - `(targetattr = "usercertificate")(version 3.0;acl "selfservice:Users can manage their own X.509 certificates";allow (write) userdn = "ldap:///self";)`

redhat®

THANK YOU!

linkedin.com/company/red-hat

twitter.com/RedHatNews

facebook.com/redhatinc

plus.google.com/+RedHat

youtube.com/redhat