POLE STAR
www.polestarglobal.com

Purplefinder Enterprise Platform Security with LDAP

Peter Potts

27th October 2010

Resources

- Manning Book: LDAP Programming, Management and Integration
- Apache Documentation & download: http://directory.apache.org
- PEP R2 LDAP Example: http://repository.enterprise.purplefinder.com
- Available on public Maven repositories.

Directory Service

- A directory service is a system that stores and provides access to information in a directory.
- A directory is a map between names and values.

Example 1: A telephone directory is map from peoples names to telephone numbers.

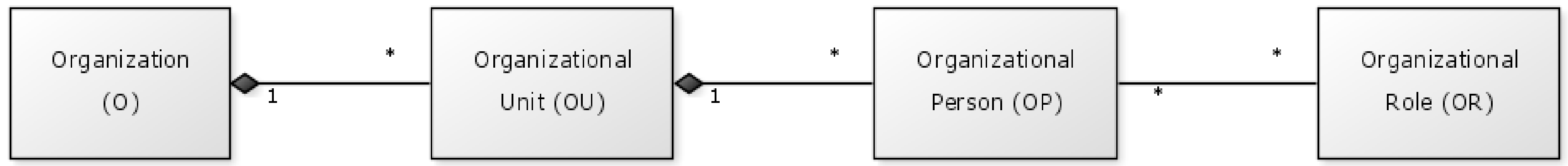
Example 2: Domain Name System (DNS) is a hierarchical naming system for computers and services connected to the internet.

POLE STAR
www.polestarglobal.com

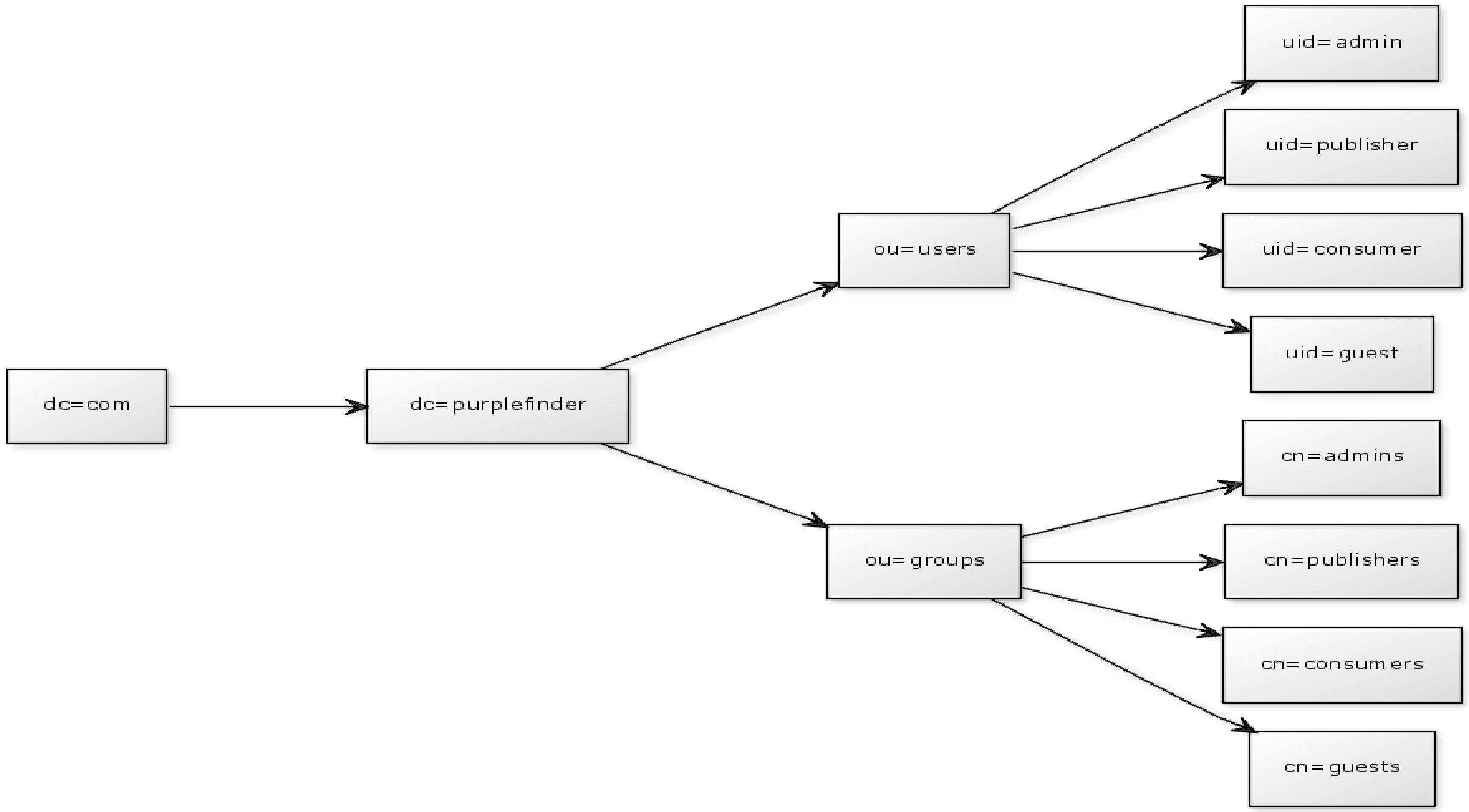
X.500 Directory Service

- A standard way to develop an electronic directory of people and devices in an organization.
- There is a single Directory Information Tree (DIT), a hierarchical organization of entries which is distributed across one or more servers, called Directory System Agents (DSA).
- An entry consists of a set of attributes, each attribute with one or more values.
- Each entry has a unique Distinguished Name (DN).

Organogram


Organization (O)
1
*
Organizational Unit (OU)
1
*
Organizational Person (OP)
*
*
Organizational Role (OR)

Directory Information Tree



Standard Schemas

- There are industry-standard schemas defined in RFC 2256 by IETF.
- Standard object classes
- Standard attribute types

Example: objectClass=organization

Superior: top

Required: o

Allowed: userPassword, postalAddress, l

Organizational Person and Group

```
dn → uid=potts,ou=users,dc=purplefinder,dc=com
      objectClass → top
      objectClass → inetOrgPerson
      cn → Peter Potts
      sn → Potts
      uid → potts
      userPassword → {SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g=

dn → cn=Research,ou=groups,dc=purplefinder,dc=com
      objectClass → top
      objectClass → groupOfNames
      cn → Research
      member → uid=potts,ou=users,dc=purplefinder,dc=com
      member → uid=ozkan,ou=users,dc=purplefinder,dc=com
```

Password Digester

```
case class PasswordDigester(algorithm: String, password: String) {
  val digestedPassword = {
    val messageDigest = MessageDigest getInstance algorithm
    messageDigest.update(StringTools getBytesUtf8 password)
    val base64Encoder = new BASE64Encoder
    val encodedPassword = base64Encoder.encode(messageDigest.digest)
    String.format("{%s}%s", algorithm, encodedPassword)
  }
}
// SHA = Secure Hash Algorithm
class PasswordDigesterSpec extends SpecificationWithJUnit {
  "A password digester" should {
    "calculate the SHA hash of a password" in {
      val target = new PasswordDigester("SHA", "password")
      target.digestedPassword mustEqual "{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g="
    }
}
```

- Run com.purplefinder.enterprise.r2.ldapexample. PasswordDigester.

LDAP

- Lightweight Directory Access Protocol (LDAP)
- Industry standard directory access protocol.
- An IP based application protocol for querying and modifying the data of directory services.
- Latest version of LDAP is Version 3.
- Specified by Internet Engineering Task Force (IETF).
- LDAP URL
  ldaps://beaker.london.purplefinder.com:636

Security

- Authentication, authorization, privacy, availability and integrity.
- Authentication is the process of proving identity.
- Authorization is all about access control rules (ACL). It answers the question: Does entity w have access to perform action x on resource y if condition z is met?

Security with LDAP

- LDAP bind operation is the crux of authentication.
- DN and user password are typical credentials.
- User password should not exist in plain text.
- User password should be hashed at all times.
- Use TLS connection for privacy.
- Authorization is achieved by using groups or roles.
- Roles are groups with an implicit set of permissions.

Groups and Roles

- Group

```
objectClass=groupOfNames
cn=<Name of group>
member=<DN of a group member>
member=<DN of another group member>
```

- Role

```
objectClass=organizationalRole
cn=<Name of role>
roleOccupant=<DN of a role-filler>
roleOccupant=<DN of another role-filler>
```

Microsoft Active Directory

- Active Directory is a Directory Service with LDAP.
- Includes industry standard schemas.
- But uses highly properietory schemas too.
- Access with Apache Directory Studio.

Apache Directory Server (ApacheDS)

- ApacheDS is an embeddable directory server entirely written in Java.
- It has been certified LDAPv3 compatible by the Open Group.
- Run `com.purplefinder.enterprise.r2.ldapexample.ArchetypalLdapServer`.
- Access with Apache Directory Studio.

“LDAP” is not a relational database

- Optimized for queries but very slow for updates.
- No support for relational integrity.
- No transactions.
- Supports multi-valued data fields.
- Excellent as a central repository for very slowly changing information required by a loosely coupled set of applications.
- Such as users, passwords, organizations, roles, ships and mobile terminals.

JNDI includes LDAP

- Java Naming and Directory Interface (JNDI).
- JNDI is part of Java EE.
- No special drivers required for LDAP.
- Standard classes found in javax.naming.directory package.

POLE STAR
www.polestarglobal.com

Change Password Using LDAP

```
val dn = "uid=consumer,ou=users,dc=purplefinder,dc=com"
val oldPassword = PasswordDigester("SHA", "password").digestedPassword
val newPassword = PasswordDigester("SHA", "secret").digestedPassword

val properties = new Properties {
    put(Context.INITIAL_CONTEXT_FACTORY, classOf[LdapCtxFactory].getName)
    put(Context.PROVIDER_URL, "ldap://localhost:10389")
    put(Context.SECURITY_AUTHENTICATION, "simple");
    put(Context.SECURITY_PRINCIPAL, dn);
    put(Context.SECURITY_CREDENTIALS, oldPassword);
}

val context = new InitialDirContext(properties)
val attr = new BasicAttribute(USER_PASSWORD_AT, StringTools getBytesUtf8 newPassword)
val mods = Array(new ModificationItem(DirContext.REPLACE_ATTRIBUTE, attr))
context.modifyAttributes(dn, mods)
context.close
```

Run `com.purplefinder.enterprise.r2.ldapexample.ChangePasswordApplication`.

Embedded Directory Service

```
val directoryService = new DefaultDirectoryService {
  getChangeLog.setEnabled(false)
  setDenormalizeOpAttrsEnabled(true)
  setWorkingDirectory(createWorkingDirectory)
  addPartition(createPartition)
}
directoryService.startup
  directoryService.loadDirectoryInformationTree(directoryInformationTree)
  val ldapServer = new LdapServer {
    setDirectoryService(directoryService)
    setAllowAnonymousAccess(true)
    setTransports(Array(new TcpTransport(portNumber)): _*)
  }
  ldapServer.start
    printf("Press enter to quit: ")
    readLine
  ldapServer.stop
directoryService.shutdown
```

Directory Information Tree

```
val archetypal = DirectoryInformationTree(
   "enterprise",
   "dc=enterprise,dc=purplefinder,dc=com",
   List(
     "" --> (
             OBJECT_CLASS_AT --> (TOP_OC, DOMAIN_OC, EXTENSIBLE_OBJECT_OC),
             "dc" --> "enterprise"),
     "ou=users" --> (
             OBJECT_CLASS_AT --> (TOP_OC, ORGANIZATIONAL_UNIT_OC),
             OU_AT --> "users"),
     "uid=admin,ou=users" --> (
             OBJECT_CLASS_AT --> (TOP_OC, INET_ORG_PERSON_OC),
             CN_AT --> "Horacio Nelson",
             SN_AT --> "Nelson",
             UID_AT --> "admin",
             USER_PASSWORD_AT --> "{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g="),
    ...
     "cn=publishers,ou=groups" --> (
             OBJECT_CLASS_AT --> (TOP_OC, GROUP_OF_NAMES_OC),
             CN_AT --> "publishers",
             MEMBER_AT --> ("uid=admin,ou=users", "uid=publisher,ou=users")),
```

JAAS

- Java Authentication and Authorization Service.
- JAAS is part of Java SE.
- Default configuration file is login.config in the resources folder.
- This configuration file allows security to be configured for as many domains of operation as required.
- For example, access control for ActiveMQ queues might be considered a domain of operation.

JAAS for ActiveMQ

- Authentication: A user name and password is provided when a JMS client connects to an ActiveMQ broker.
- Authorization: The send, receive & admin rights of queues and topics are controlled with users and groups.

POLE STAR
www.polestarglobal.com

JAAS with Properties for ActiveMQ

- Run com.purplefinder.enterprise.r2.ldapexample.PropertiesAuthenticationIntegrationSpec
- login.config

```
activemq-properties-domain {
    org.apache.activemq.jaas.PropertiesLoginModule required
        org.apache.activemq.jaas.properties.user="users.properties"
        org.apache.activemq.jaas.properties.group="groups.properties”;
};
```

- users.properties

```
admin=password
publisher=password
consumer=password
guest=password
```

- groups.properties

```
admins=admin
publishers=admin,publisher
consumers=admin,publisher,consumer
guests=guest
```

Authorization Map Entries

```
new AuthorizationEntry {
        setQueue(">")
        setRead(admins)
        setWrite(admins)
        setAdmin(admins)
},
new AuthorizationEntry {
        setQueue("queues.>")
        setRead(consumers)
        setWrite(publishers)
        setAdmin(publishers)
},
new AuthorizationEntry {
        setQueue("queues.demo")
        setRead(guests)
        setWrite(guests)
},
new AuthorizationEntry {
        setTopic("ActiveMQ.Advisory.>")
        setRead(List(admins, publishers, consumers, guests).mkString(","))
        setWrite(List(admins, publishers, consumers, guests).mkString(","))
        setAdmin(List(admins, publishers, consumers, guests).mkString(","))
}
```

JAAS with LDAP for ActiveMQ

- Run com.purplefinder.enterprise.r2.ldapexample.ApacheDSAuthenticationIntegrationSpec
- login.config

```
activemq-apacheds-domain {
    org.apache.activemq.jaas.LDAPLoginModule required
        initialContextFactory=com.sun.jndi.ldap.LdapCtxFactory
        connectionURL="ldap://localhost:10389"
        connectionUsername="uid=admin,ou=system"
        connectionPassword="secret"
        connectionProtocol="s"
        authentication="simple"
        userBase="ou=users,dc=enterprise,dc=purplefinder,dc=com"
        userRoleName="dummyUserRoleName"
        userSearchMatching="(uid={0})"
        userSearchSubtree=false
        roleBase="ou=groups,dc=enterprise,dc=purplefinder,dc=com"
        roleName="cn"
        roleSearchMatching="(member={0})"
        roleSearchSubtree=false;
};
```

POLE STAR
www.polestarglobal.com

Active Directory Application

- Run com.purplefinder.enterprise.r2.ldapexample.ActiveDirectoryApplication
- login.config

```
activemq-active-directory-domain {
    org.apache.activemq.jaas.LDAPLoginModule required
        initialContextFactory=com.sun.jndi.ldap.LdapCtxFactory
        connectionURL="ldap://beaker.london.purplefinder.com:389" // Use TLS
        connectionUsername="CN=Peter Potts,OU=Pole Star Users,DC=london,DC=purplefinder,DC=com"
        connectionPassword="bogus"
        connectionProtocol="s"
        authentication="simple"
        userBase="OU=Pole Star Users,DC=london,DC=purplefinder,DC=com"
        userRoleName="memberOf"
        userSearchMatching="(userPrincipalName={0})"
        userSearchSubtree=false
        roleBase="OU=Pole Star Groups,DC=london,DC=purplefinder,DC=com"
        roleName="CN"
        roleSearchMatching="(member={0})"
        roleSearchSubtree=false;
};
```