Intro to LDAP

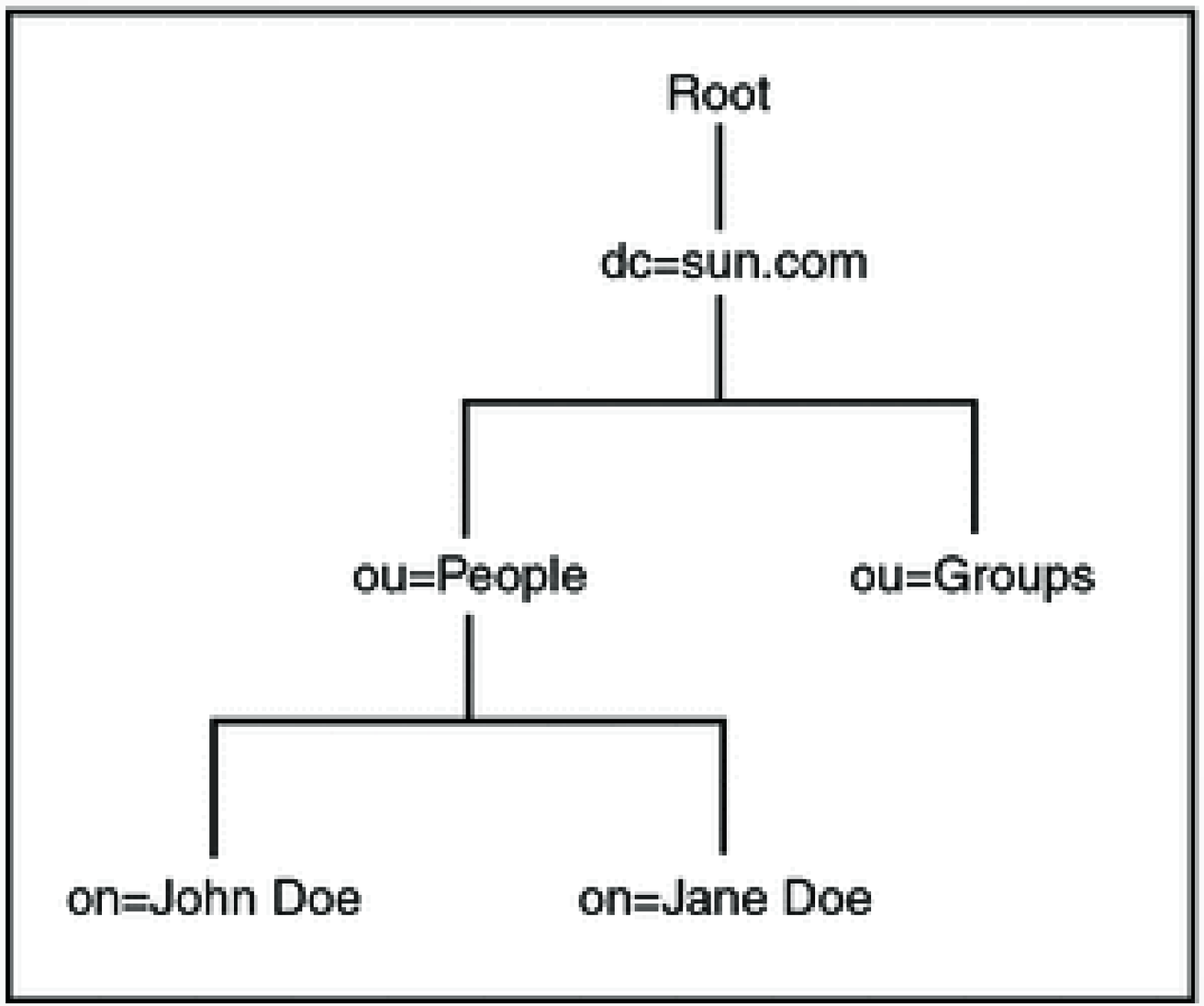
What is LDAP?

- Lightweight Directory Access Protocol
- Tcp based
- Server, client design
- In a nutshell: LDAP is the protocol that interacts with data contained in directory servers
- OpenLDAP (unix) and Active Directory (Microsoft) implement LDAP, JNDI (Java client API), ldaptor (Python server+client implementation)

LDAP Directory Server

- Similar to an SQL database but with no relational tables
- Data is represented as tress of enteries

LDAP Directory Server



LDAP Usecase

- It can be used potentially anywhere a relational DB is used, though it's primarily used for account info retreval and authentication
- Excels in any "write/update once, read/query many times" task
- Personal use: LDAP is used to keep consistent users between all my linux machines
  - ^^ Demo this

LDAP Queries

- Client cli tool: ldapsearch
- Idea: Query down the tree until a leaf is hit, then return its contents
- In the example above, the query cn=John doe ou=People, dc=Sun, dc=com
- Ou, dc, cn are called attributes
- These can be custom, but those are some of the most common ones used, to modify the schema must be altered
- ObjectClass → a “meta attribute”

LDAP Queries

- cn=John doe ou=People, dc=Sun, dc=com is a distingusihed name (DN) as it has a path to the object from root
- Each DN must be unique
- Realtive Dns (RDN) ex: cn=John Doe, ou=People
- What is retrived is an LDIF file (LDAP Data Interchange Format) which is a simple plaintext file

LDAP Usecase: User sync

- Create a user + LDAP entry on server that holds the uid/gid etc for the created user
- Let the client retrieve this data to create the same user
- Demo

LDAP Injection

- Malcious queries can be constructed similar to SQL injection
- LDAP sytnax: (query) (filter)
- Both query and filter have syntax:
  - (Attribute operator value)
- A filter reduces the values returned from the query
- Operators include =, >=, <=, etc..

  Multiple queries can be combined using boolean ops such as & and | etc
  - Ldapsearch supports both normal and Polish notation, for the tutorial only Polish notation is supported

LDAP Injection – Known info

- General injection approach:
  - Try to close the orignal query by ending the string, adding) and combine it with a or'd true statement
- For example, imagine a client forms this search
  - (&(user=$1)(password=$2))
  - $1 = "bob)(|(user=bob"
  - $2 = "aaa)"
  - The server would see the following query:
    - (&(user=bob)(|(user=bob)(pass=aaa)))
      - T and (T or F) = T

LDAP Injection – Blind

- What if we want to see the attribute data?
  - Use bruteforce + wildcards! → Guess the attributes with value * and build a string
  - Lets say we have a client that outputs a telephone number given that the user is of objectclass person
  - (&(user=$1)(objectClass=Person))
    - In our attack we let, $1= ”bob)(userPassword=a*"
    - This creates the query (&(uid=bob)(userPassword=a*)(objectClass=person))
    - If this is true, then we know that the first character of attribute userPassword is a, otherwise try b, etc..
    - Repeat this process until the whole string is constructed