

Homework 4

LDAP

csjhuang

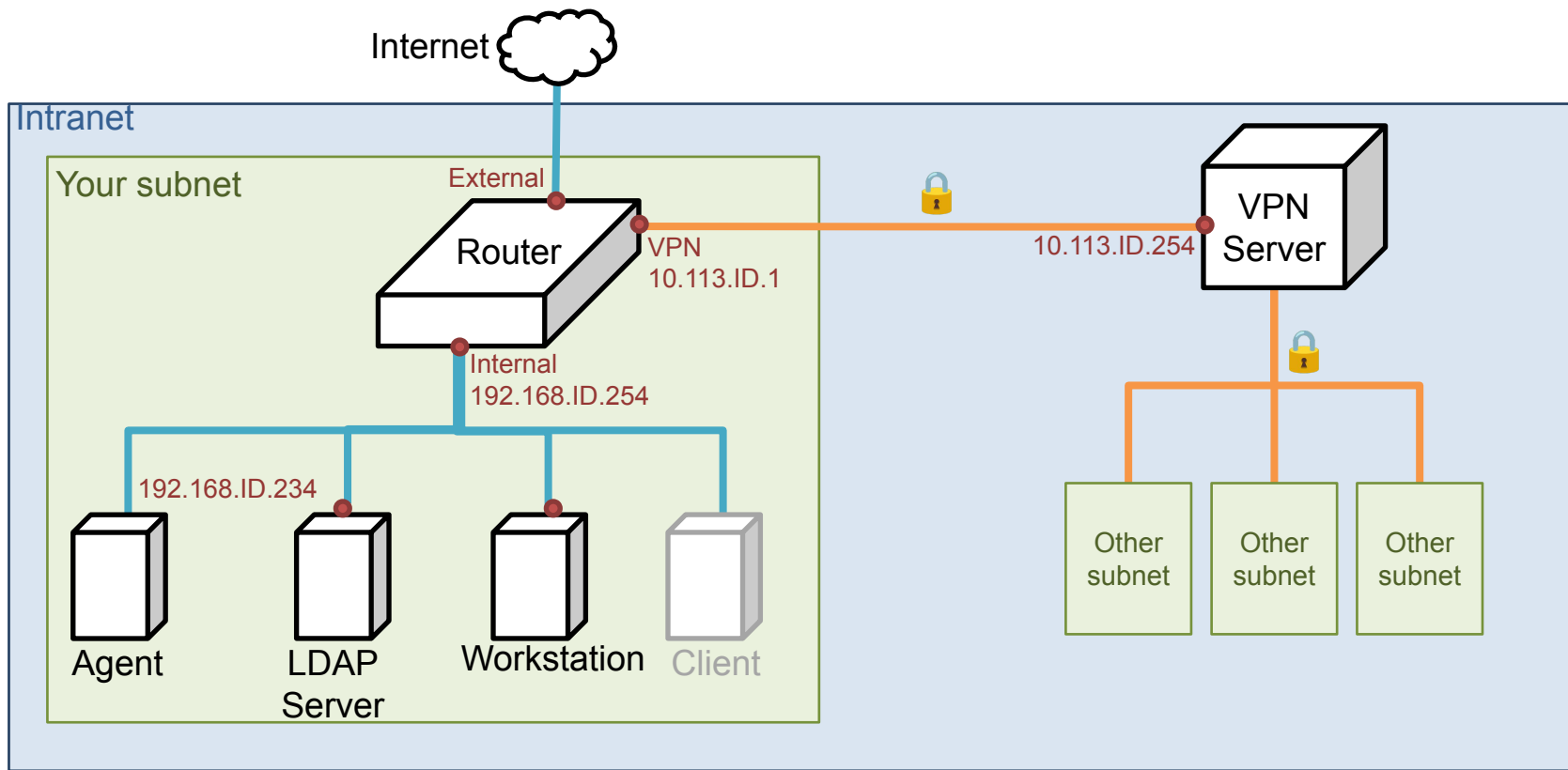
國立陽明交通大學資工系資訊中心

Computer Center of Department of Computer Science, NYCU

Purposes

- Build a basic LDAP service
- Understand how to...
 - configure LDAP server
 - manage LDAP data using LDIF
 - auth and permission control on Unix client with LDAP server
 - customize your own objectClass and using OLC(on-line configuration)

Overview - Architecture



Overview (cont.)

- A simple LDAP server
 - LDAP client
- One or more Workstations
 - LDAP client

Requirements

- LDAP Server
 - IP: 192.168.ID.y/24 with static DHCP, where y is arbitrary.
 - Hostname: ldap.{ID}.nasa. (5%)
 - Base DN: dc=<ID>, dc=nasa
 - LDAPS and force TLS search (8%)
 - Not LDAP over TLS (StartTLS) (2%)
 - Use certificate generator to get your key and certificate

Requirements

- Organizational Unit Naming
 - People
 - Group (posixGroup)
 - Ppolicy
 - SUDOers
 - Fortune (our customize objectClass)

Requirements

- Workstation
 - IP: 192.168.ID.y/24 with static DHCP, where y is arbitrary
 - Hostname: workstation.{ID}.nasa. (5%)

Requirements

We need two posix group in LDAP:

- ta group (GID=10000)
 - can login (ssh) into LDAP server and any workstations (6%)
 - can use sudo for any command (7%)
 - ex. `sudo adduser`
- stu group (GID=20000)
 - can login (ssh) into workstations, cannot login into LDAP server (6%)
 - only allow sudo for `ls` command (7%)
- You need use “LDAP” to implement above requirements
 - Including sudo rules and ssh key!
- TA will add any named user using generalta into these group (10%)

Requirements

Add an user with DN “uid=generalta,ou=People,<Base DN>”

- This user under ta group, use ta group permission
- Allow this user to connect via SSH with both ssh public key and password
 - uid: generalta
 - uid number: 10000
 - public key: <ta’s public key> # See p.12
 - user password: <your TA_PASSWORD> # Same as HW3
 - user password need hash

Requirements

TA's public key: <https://nasa.cs.nycu.edu.tw/na/2024/slides/hw4.pub>

- Public key:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJvrjeKMDtDZVVQCUY/hpVYP4jp9+WUwKq984TKgRp6G 2024-na-hw4
```

- User can set their authorized keys with the `sshPublicKey` attribute

Requirements

Add another user with DN “uid=stu<ID>,ou=People,<Base DN>”

- This user under stu group, use stu group permission
- Allow this user to connect via SSH with both ssh key and password
 - uid: stu<ID>
 - e.g. stu1, stu55
 - uid number: 20000 + <ID>
 - e.g. 20001, 20055
 - user password: <your TA_PASSWORD>

Requirements

- Configure LDAP Client on every machine
 - Configure LDAP for login (ssh) authentication
 - can use password or public key to login
 - When you add a user into LDAP, this user can login on any workstation or LDAP Server
 - Login permissions at Page 7

Requirements

- Set proper LDAP access control
 - Allow generalta to manage users and groups
 - Allow every users to modify their own userPassword, loginShell and sshPublicKey (6%)
 - Set other attributes as read-only (6%)
 - Allow users to search all user data but other users' password (6%)
 - i.e., users can only read their own password
 - generalta can write to it but not read!

Requirements

- Set password policy for each user (10%)
 - userPassword can't same as previous when change password
 - But can set password as previous two time used
 - You need implement this by LDAP way
 - password requires at least 8 characters long
 - password must contains at least 3 different classes of characters:
 - Upper-case characters
 - Lower-case characters
 - Digits
 - Special characters
 - Hint: ppolicy overlay & pwdCheckModule

Requirements

- Add an OU(Fortune) that contains fortunes (4%)
- Add an ObjectClass fortune with **on-line configuration (OLC)** (6%)
 - **schema**
 - objectClass's oid should be under the [UUID branch](#)
 - extend from **top**, add **author** field(octetString), and **id** field(integer)
 - **author's** matching and order should be “**case insensitive, space insensitive**”
 - use existing **description**([RFC 4519](#)) attribute to place sentences
 - we would check whether this objectclass is in database (cn=cofnig)

Requirements

- Import fortunes (4%)
 - from given csv file ([link](#))
 - 3 column
 - ID
 - Authour
 - Description
- enable features (4%)
 - server side sorting
 - pagination
 - Hint: slapd-sssylv

```
ID,Author,Description
1,Richard Feynman,The first principle is that you must
not fool yourself -- and you are the easiest person to
fool.
```

```
dn: cn=fortune-1,ou=Fortune,dc=254,dc=nasa
objectClass: fortune
objectClass: top
cn: fortune-1
author: Richard Feynman
id: 1
description: The first principle is that
you must not fool yourself -- and you
are the easiest person to fool.
```


Attention

- Your work will be tested by our online judge system
 - Submit a judge request when you are ready.
 - You can submit request multiple times. However, the score of the latest submission instead of the submission with the highest score, will be taken.
 - Late submissions are not accepted.
 - Please check your score at OJ after judge completed.
- Scoring starts at : **2024/5/23 (Thur.) 21:00**
 - The cool-down time is 15 minutes
- Deadline: **2022/6/5 (Wed.) 23:59**

Help

- Questions about this homework.
 - Make sure you have studied through lecture slides and the HW spec.
 - Clarify your problems and search it to find out solutions first.
 - Ask them on <https://groups.google.com/g/nctunasa> .
 - Be sure to include all the information you think others would need
- Do not mail us unless it's personal or you're making an appointment.

Good Luck!

國立陽明交通大學資工系資訊中心

Computer Center of Department of Computer Science, NYCU