

IM3014 Linux System Administration Practice

Homework 3: Enterprise Ops

Introduction

You are a junior system administrator at **IM Corp.** The company has recently grown—adding more servers and expanding into new business sectors—making it harder for your team to manage hardware and software. Manually configuring environments on a per-machine basis is no longer sustainable. Your manager has assigned two tasks:

1. Set up an LDAP directory service to centralize user credentials and simplify account management.
2. Set up a private APT repository to enable internal teams to share and distribute their packages.

By the end of this assignment, you will have configured two essential tools for Linux system administration, deepening your ability to support a large user base in an organization.

Problem Specification

LDAP Directory Service

In this section, you will configure a local LDAP service on your VM, add groups and users, secure it with LDAPS, and connect from your personal computer.

1. Install Required Packages

Install OpenLDAP server and tools (e.g., `slapd`, `ldap-utils` on Debian/Ubuntu).

2. Initial Configuration

Reconfigure the service so that:

- **Domain:** `im.ntu.edu.tw` ⇒ base DN `dc=im,dc=ntu,dc=edu,dc=tw`.
- **Organization:** set to `<your-student-id>`.

Verification: `slapcat` shows the base DN under the main database.

3. Create Organizational Units

Create `ou=People` and `ou=Groups` under the base DN.

Verification:

```
$ ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(objectClass=organizationalUnit)" ou
```

4. Create Users and Groups

Create two groups: `eng` and `intern`. Then create three user accounts using *your* first name, last name, and student ID.

- Example (TA): first name = `An-Che`, last name = `Liang`, student ID = `b11705009`.
Create users: `an-che`, `liang`, `b11705009`.
- Place the `an-che` and `liang` accounts in group `eng`.
- Place the `b11705009` account in group `intern`.

Verification:

```
$ ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(uid=<your-uid>)" uid,uidNumber,gidNumber
$ ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(cn=eng)" cn gidNumber memberUid
```

5. Generate a Certificate Authority (CA)

Generate `ca.key` and `ca.crt`. Keep `ca.key` private.

Verification: paste the contents of `ca.crt`

6. Generate Server Key & CSR

Generate `server.key` and `server.csr` for your LDAP host (ensure the CSR includes a SAN or CN matching your server hostname).

Verification: paste the contents of `server.csr`.

7. Sign the Server Certificate

Use your CA to sign `server.csr` and produce `server.crt`.

Verification: paste the contents of `server.crt`.

8. Enable LDAPS and Trust Your CA Locally

Configure `slapd` to use `server.crt`, `server.key`, and `ca.crt`; enable LDAPS (`ldaps://`) and update the local trust store so LDAP tools trust your CA.

Verification:

```
$ ldapsearch -x -H ldaps://<domain-of-the-VM>:<assigned-port-of-the-VM> \
-b "dc=im,dc=ntu,dc=edu,dc=tw" -s base namingContexts
```

9. GUI Verification (Apache Directory Studio)

Install Apache Directory Studio on your personal computer. Connect to your VM via the forwarded ports:

- TCP 389 is forwarded to 389XY
- TCP 636 is forwarded to 636XY

where `stnXY` is your VM ID. Use LDAPS for the connection and trust your CA.

Verification: include a screenshot of Apache Directory Studio showing a successful bind and a tree view of your **People** and **Groups** OUs.

Custom APT Repository

In this section, you will implement a small C/C++ library that serves one of the IM Corp teams, package it as a Debian (.deb) package, and publish it via a custom APT repository with GPG key pair. You will then install the package from your repository on your local machine to verify end-to-end delivery.

Your library must provide a command-line interface (CLI) that accepts arguments (e.g., input files and output paths) and behaves like a standard terminal tool. It must also include a help option (-h or --help) that prints usage information.

Library options Select exactly one library based on your VM ID: if your VM ID is `stnXY`, compute $XY \bmod 5$. For example, `stn56` $\Rightarrow 56 \bmod 5 = 1$, so you must implement the **Compression** library.

ID	Library	Description / Team
0	Path Finding	C/C++ program that reads a maze (text format) and outputs the shortest path from start to goal using BFS. (Robotics Team)
1	Compression	C/C++ program that compresses and decompresses plain ASCII files using Huffman coding. (Networks Team)
2	Statistics	C/C++ program that fits linear regression (multiple independent variables, one dependent variable). (Business Intelligence Team)
3	Image Processing	C/C++ program that applies a Gaussian blur to BMP (bitmap) images. (Camera Team)
4	Encryption	C/C++ program that encrypts and decrypts plain ASCII files with RSA. (Cyber Security Team)

Table 1: Selectable libraries and their target teams.

Provide a clear, step-by-step guide that explains your project's structure, how to package it as a Debian (.deb) package, how to set up the APT repository, and how to install the package on your local machine. To avoid naming collisions, prefix the library name with your student ID. For example, if your student ID is `b11705009` and you implement the image processing library, name it `b11705009-image-processing`.

Deliverables

You are required to setup the custom APT repository on your virtual machine during the designated time period (from 2025-10-27 to 2025-11-03) and submit the following materials:

- 1. Source Code:** Submit the complete source code and build configuration files (e.g., `Makefile` or `CMakeLists.txt`) for the custom library you implemented. The implementation must be in C or C++ only; refrain from using third-party libraries.
- 2. System Setup Report:** Write a report (maximum length: 8 pages) that documents:
 - how you setup the LDAP directory service;
 - how you setup the custom APT repository;
 - screenshots or command outputs that demonstrate the system running correctly.

Your submission should be concise, technically clear, and sufficient to allow another system administrator to reproduce your setup.