UNIT 4: LINUX ADMIN(UBUNTU)

CS – 22: Operating Systems Concepts With Unix / Linux

- Creating Linux User Account and Password
- Installing and Managing Samba Server
- Installing and Managing Apache Server
- Optimizing LDAP Services
- Optimizing DNS Services
- Optimizing FTP Services
- Optimizing Web Services
- Configure Ubuntu's Built-In Firewall
- Working with WINE

Assignment Questions

1. What is samba server?
2. LDAP Stands for _____.
3. Which command is use to create a new user in Linux?
4. What is DNS?
5. Explain web services.
6. Explain FTP optimization.
7. What is WINE?
8. Explain firewall and how to enable firewall in Linux.
9. What is samba server? Explain the installation process.
10. Explain Apache server in detail with installation

PREPARED BY: PROF. N.K.PANDYA (PHD*, MCA, BCA, BPP)
KAMANI SCIENCE COLLEGE, AMRELI(BCA/BSC)

Creating Linux User Account and Password

Create a user

The simple format for this command is useradd [options] USERNAME.

For example useradd test (as the root user - prefix with sudo if you are not logged in as root).

This will create a user called test, but it's a limited operation and will not create other useful things like their home directory or password!

Add a password (syntax as unit2)

You then add a password for the test user by using the passwd command: passwd test. This will prompt you to enter a password for the user.

There is an option for adding an encrypted password via the -p option on useradd, but this is not recommended for security purposes.

Note that the -p option doesn't allow you to input a plaintext password, it expects you to encrypt it first. This is intentionally difficult, because you should not do it! Just use the passwd command.

Installing and Managing Samba Server

A Samba file server enables file sharing across different operating systems over a network. It lets you access your desktop files from a laptop and share files with Windows and macOS users.

Step 1: Install Samba

sudo apt update

sudo apt install samba

Now that Samba is installed, we need to create a directory for it to share:

Step 2: Create sharable folder

mkdir /home/<username>/sambashare/

The command above creates a new folder **sambashare** in our home directory which we will share later.

Step 3: add folder to the config file

To add the new directory as a share, we edit the file by running:

sudo nano /etc/samba/smb.conf

The configuration file for Samba is located at **/etc/samba/smb.conf.**

At the bottom of the file, add the following lines:

[sambashare] <foldername>

comment = Samba on Ubuntu <Some text to identify folder>

path = /home/username/sambashare <path to the folder>

read only = no <yes/no>

browsable = yes <yes/no>

Then press Ctrl-O to save and Ctrl-X to exit from the nano text editor.

comment: A brief description of the share.

path: The directory of our share.

read only: Permission to modify the contents of the share folder is only granted when the value of this directive is no.

browsable: When set to yes, file managers such as Ubuntu's default file manager will list this share under "Network" (it could also appear as browseable).

Step 4: Restart samba server

Now that we have our new share configured, save it and restart Samba for it to take effect:

sudo service smbd restart

Step 5: Add/upada firewall rules

Update the firewall rules to allow Samba traffic:

sudo ufw allow samba

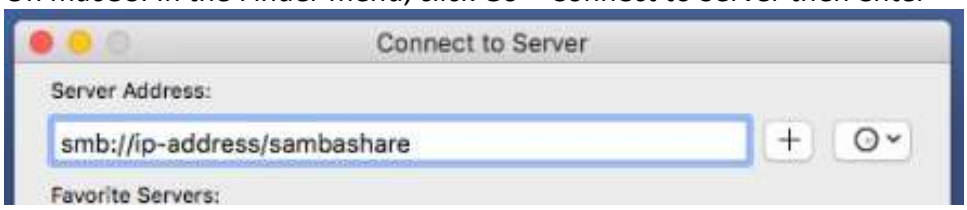
Step 6: Setting up User Accounts and Connecting to Share

Since Samba doesn't use the system account password, we need to set up a Samba password for our user account:

sudo smbpasswd -a username

Step 7: Access the folder

On macOS: In the Finder menu, click Go > Connect to Server then enter



On Windows, open up File Manager and edit the file path to:

\\ip-address\sambashare

Note: ip-address is the Samba server IP address and sambashare is the name of the share.

Installing and Managing Apache Server

Apache is an open source web server that's available for Linux servers free of charge.

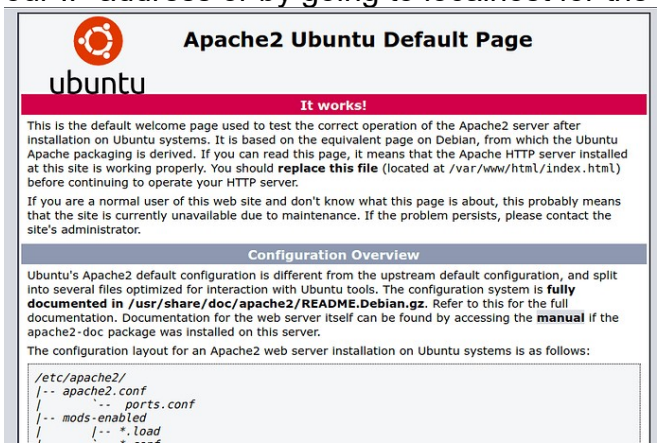
Step 1: To install Apache, install the latest meta-package apache2 by running:

sudo apt update

sudo apt install apache2

Step 2: Check if the server works

After letting the command run, all required packages are installed and we can test it out by typing in our IP address or by going to localhost for the web server.



If you see the page above, it means that Apache has been successfully installed on your server! Let's move on.

Step 3: Creating/add Your Own Website

By default, Apache comes with a basic site (the one that we saw in the previous step) enabled.
We can modify its content in /var/www/html or settings by editing its Virtual Host file found in /**etc/apache2/sites-enabled/000-default.conf.**

We can modify how Apache handles incoming requests and have multiple sites running on the same server by **editing its Virtual Hosts file**.

we're going to leave the default Apache virtual host configuration pointing to www.example.com and

set up our own at **ksc.example.com**.

Step 4:
So let's start by creating a folder for our new website in **/var/www/** by running
sudo mkdir /var/www/ksc/

We have it named ksc here but any name will work, as long as we point to it in the virtual hosts configuration file later. Now that we have a directory created for our site, lets have an HTML file in it. Let's go into our newly created directory and create one by typing:

cd /var/www/ksc/
nano index.html
Paste the following code in the index.html file:

```
<html>
<head>
  <title> Example Website : KSC</title>
</head>
<body>
  <p> I'm running this website on an Ubuntu Server server!
</body>
</html>
```

Step 5: Setting up the VirtualHost Configuration File

We start this step by going into the configuration files directory:
cd /etc/apache2/sites-available/
Since Apache came with a default VirtualHost file, let's use that as a base. (ksc.conf is used here to match our subdomain name):

cp 000-default.conf ksc.conf
Now edit the configuration file:
nano ksc.conf
We should have our email in **ServerAdmin** so users can reach you in case Apache experiences any error:
ServerAdmin yourname@example.com
We also want the DocumentRoot directive to point to the directory our site files are hosted on:
DocumentRoot /var/www/ksc/

The default file doesn't come with a **ServerName** directive so we'll have to add and define it by adding this line below the last directive:
ServerName ksc.example.com
This ensures people reach the right site instead of the default one when they type in ksc.example.com.

Step 6: Activating VirtualHost file

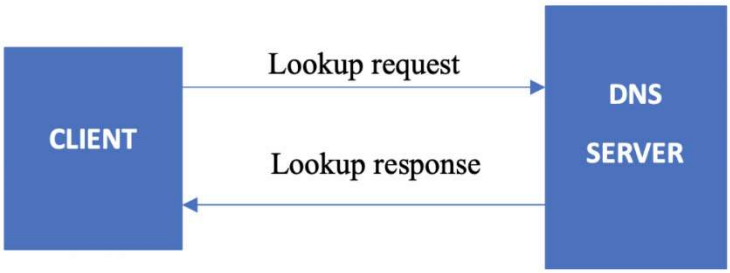
After setting up our website, we need to activate the virtual hosts configuration file to enable it. We do that by running the following command in the configuration file directory:
sudo a2ensite ksc.conf
You should see the following output
Enabling site ksc.
To activate the new configuration, you need to run:
service apache2 reload

Optimizing LDAP Services

We know that Linux keeps registered users on /etc/passwd file, so if you want to access the machine, you must have a user on that file. If you are working with one or few machines, that should be OK, but what if you have hundreds of machines or maybe thousands, and how you will maintain user management tasks like password modification or any other administrative task like somebody left the work and you need to close his account, would you go to every machine to do that?
That could be a nightmare, or you need to create a new account. In this case, we need a centralized user account management system, a database to keep all information related to user accounts. The most used solution for this problem is the Lightweight Directory Access Protocol (LDAP). LDAP not only keeps a list of users, but you can also use it as storage for your files. You can use it for authenticating users as we mentioned above. Also, you can store DNS records in the LDAP server. Another usage for LDAP, you can use it as a yellow pages directory service for an organization to provide information about users or employees, departments, contact information, phone numbers, addresses, private data, or whatever.

Optimizing DNS Services

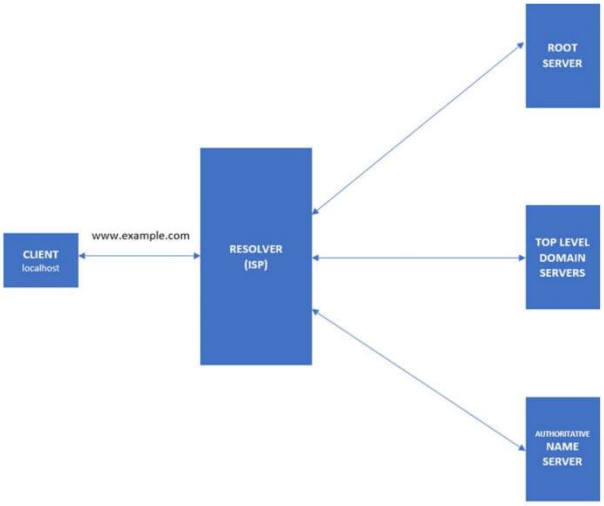
The Domain Name System (DNS) is used to resolve (translate) hostnames to internet protocol (IP) addresses and vice versa. A DNS server, also known as a nameserver, maps IP addresses to hostnames or domain names.
How DNS works
When a client requests information from a nameserver, it usually connects to port 53, and then the nameserver resolves the name requested.



- Sending a request from the DNS client to the DNS server is called a lookup request.
- Getting a response from the DNS server to the DNS client is called a lookup response.
- The system on which the DNS service is configured is called a DNS server.
- The system that accesses the DNS server is called a DNS client

Where does DNS get IP addresses?

You might wonder how DNS gets the IP of the corresponding hostname or domain name. How does DNS search among different IP addresses and associate your domain name correctly? Who stores those mappings between domain names and IP addresses? The DNS workflow illustrates how communication happens within DNS and how it resolves the addresses.



- When the client searches for the domain www.example.com, the request will initially go to the internet service provider's (ISP) resolver. It will respond to the user's request to resolve a domain name.
- If the IP address is not found on the resolver, the request is forwarded to a root DNS server and later to the top-level domain (TLD) servers.
- TLD servers store information for top-level domains, such as **.com** or **.net**.
- Requests are forwarded to the nameservers, which know detailed information about domains and IP addresses.
- Nameservers respond to the ISP's resolver, and then the resolver responds to the client with the requested IP.
- When the resolver doesn't know the IP, it stores the IP and its domain in a cache to service future queries.

Install and configure DNS: BIND is a nameserver service responsible for performing domain-name-to-IP conversion on Linux-based DNS servers.

sudo install bind

The BIND package provides the named service. It reads the configuration from the /etc/named and /etc/named.conf files. Once this package is installed, you can start configuring DNS.

Configure the /etc/named.conf file

First, add or edit the two values in the options field. One is the DNS server address, and the other is the allow-query to any.

nano /etc/named.conf

listen-on port 53 { 127.0.0.1; 192.168.25.132; };

allow-query { localhost; any; };

Here are the values from the above file:

192.168.25.132 – DNS server address

any – matches every IP address

Optimizing FTP Services

FTP or the File Transfer Protocol is the most popular network protocol that is used to transfer files and information between two systems over a network. However, the FTP by default does not encrypt the traffic, which is not a secure method and can result in an attack on a server.
This is where VSFTPD comes which stands for Very Secure FTP Daemon and is a secure, stable, and fast FTP server. VSFTPD is licensed under GNU GPL.
For most of the Linux distributions, VSFTPD is used as a default FTP server.

Installing FTP server
Step 1: Install VSFTPD
Our first step will be to install VFTPD on our system. To do so, launch the Terminal.
Then issue the following command in the Terminal to update the system repository index:
sudo apt update
Then install VSFTPD using the following command in Terminal:
sudo apt install -y vsftpd
After the installation of VSFTPD is completed, we will move towards configuration.

Step 2: Configure VSFTPD
The VSFTPD can be configured through the /etc/vsftpd.conf file. Edit the /etc/vsftpd.conf file using the following command in Terminal:
sudo nano /etc/vsftpd
Now add or uncomment the following lines (if already added in the file):

```
listen=NO
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
chroot_local_user=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=Yes
pasv_enable=Yes
pasv_min_port=10000
pasv_max_port=10100
allow_writeable_chroot=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

Once done, save and close the /etc/vsftpd.conf file.

Step 3: Allow ports in firewall

If a firewall is running on your system, you will need to allow some FTP ports through it. Issue the following commands in Terminal to allow the ports 20 and 21:

sudo ufw allow 20/tcp

sudo ufw allow 21/tcp

You can verify whether the port has been allowed in the firewall or not using the following command in Terminal:

sudo ufw status

Step 4: Enable and run VSFTPD

Now the VSFTPD is configured and allowed in the firewall; now we can enable and run the VSFTPD services. Here are the commands to do so:

To enable the VSFTPD service to start on boot, issue the following command in Terminal:

sudo systemctl enable vsftpd.service

To run the VSFTPD service, issue the following command in Terminal:

sudo systemctl start vsftpd.service

If you need to restart the VSFTPD service after making any configuration changes, issue the following command in Terminal:

sudo systemctl restart vsftpd.service

To verify if the VSFTPD is active and running, issue the following command in Terminal:

sudo systemctl status vsftpd.service

Step 5: Create an FTP user

Next, create a user account that will be used to test the FTP connection. Issue the following commands in Terminal to create a user account and set a password:

sudo adduser <username>

sudo passwd <username>

Step 6: Test FTP connection

Now our FTP server is ready, so it's time to test the FTP connection. To test FTP connection locally, issue the following command in Terminal by replacing the <ip-address> by the actual IP address of your FTP server:

ftp <ip-address>

Optimizing Web Services

The definition of a Web Service is "a software system designed to support interoperable machine-to-machine interaction over a network." Web services (at times called application services) are services (more often than excluding some combination of programming and information, however, includes human resources also) that are made accessible from a business' Web server for Web users or other Web-connected programs. Providers of Web services are by and largely known as application specialist co-ops. Web services go from such real administrations as capacity administration and client relationship management (CRM) down to substantially more constrained administrations, for example, the outfitting of a stock statement and the checking of offers for an auction item. The quickening creation and accessibility of these services is a noteworthy Web trend.

Web Performance Optimization happens by checking and investigating the performance of your web application and recognizing approaches to enhance it. Web applications are a blend of server-side and customer side code. Your application can have performance issues on either side or the two should be optimized. The customer side identifies with execution as observed inside the web browser. This incorporates starting page load time, downloading all resources, JavaScript that keeps running in the program, and more. The server-side identifies with to what extent it takes to keep

running on the server to execute requests. Optimizing the execution on the server, for the most part, revolves around advancing things like database questions and other application dependencies. We should investigate how to improve execution for both the client and server sides.

Configure Ubuntu's Built-In Firewall

Check Ubuntu Firewall Status

Before disabling the UFW(Uncomplicated Firewall) firewall, it is a good idea to check its status first. In Ubuntu, the firewall is disabled by default. How do you know if your firewall is on?

To check the current status of the firewall, execute the command in your command terminal:

sudo ufw status

ufw firewall is active on this system

As we have determined the current state, now we can proceed to disable the UFW firewall.

Disable Ubuntu Firewall

A firewall is a vital element in a network and server security. However, while testing or troubleshooting, you might need to shut down or stop the firewall.

To disable the firewall on Ubuntu, enter:

sudo ufw disable

The terminal informs you that the service is no longer active.

If you disable the firewall, keep in mind that your firewall rules are still in place. Once you enable the firewall again, the same rules that were set up prior to the deactivation will apply.

Enable Firewall

Learning how to enable the firewall on Ubuntu is vital.

To enable the firewall on Ubuntu, use the command:

sudo ufw enable

As with the ‘disable’ command, the output confirms that the firewall is once again active.

Using UFW to Set Firewall Rules

UFW does not provide complete firewall functionality via its command-line interface. However, it does offer an easy way to add or remove simple rules.

A good example is opening an SSH port.

For example:

sudo ufw allow 22

Once the terminal confirms that the rule is now in place, check the status of the firewall with the ‘status’ command:

sudo ufw status

The output is going to reflect the fact that an SSH port is now open.

Reseting UFW Firewall Rules

If you need to reset all rules back to default settings, use the reset command:

sudo ufw reset

Working with WINE

Basically, Wine is a sort of software, or we can specifically say that it is a type of open-source software. It is available for everyone that means anyone can download it and use it without spending a penny. It can be considered as one of the most important softwares that we may need, specifically, if we're a Linux user, as it allows us to run windows applications on several Linux and UNIX based operating systems (or other POSIX compliant operating systems).

In simple words, we can say that if we are using a Linux or Unix-based operating system and we need to run some Windows applications for a certain task or just for entertainment. One can do so by installing the Wine Software. Wine Software allows it users to run the various windows application on the Linux based operating system without requiring any other additional software like a virtual

machine. All we need is to install the Wine software only because, With Wine's help, we never need to install windows on wer system, which a time taking process (or using a virtual box).
It may be quite possible that many users may think it is also an emulator like several other emulators available on the internet. But we are not right here because it cannot be considered as an emulator. Instead, it instantly translates Windows API calls into POSIX calls and eliminates the performance and memory penalty of other methods, and allows we to cleanly integrate Windows applications into wer desktop.

Step 1:Installing Wine software

To install the Wine software seamlessly, we will recommend installing Wine using the standard Ubuntu Repository. Through this way, we can have a more stable and reliable version on our computer.

Type the following given command in the terminal and press enter:
sudo apt install wine64
it can ask us to enter the root user password, so enter the password to continue.
While installing the Wine, we will be prompted with the y/n option, so type "y" and press enter to continue the installation process
Once the installation process gets completed, exit out the terminal using the given command.
Once we exit out the terminal, the Wine software is all set and ready to use.
Now, to verify the installation of Wine software is successfully completed or not, we can use the following command:
wine –version

After successfully downloading and installing the Wine, let us see how to use this for running window applications. All we need is the .exe file of the desired window application that we want to run.
In the given image, we can see that we have an .exe file of Notepad++, which is a window's application. To run it on our Linux operating system with the help of Wine follows the given instructions carefully:

Step1. First of all, open the terminal by pressing Ctrl+Alt+T on the keyboard.

Step2. Once the terminal gets open, navigate to the folder (or directory) where we have ".exe file." As we can see in our case, our .exe file is located on the desktop.
Step3. Type the following command to install windows application in our Ubuntu operating system.
Wine <Our window application name>
Once we press enter after typing the above command, the installation process of Notepad++ will be started:

Step4. Now follow the instruction to install the notepad++, once finish you can use the software as on the window.