MIDTERM REPORT: Linux System Security

CENG3544, COMPUTER AND NETWORK SECURITY

Beyza Kurt
beyzakurt@posta.mu.edu.tr

Tuesday 12th May, 2020

Abstract

Our main goal in this report is to learn what we can do to secure the Ubuntu operating system in many ways. We will work on Apache, OpenSSh and Vsftpd services, system structure and their security.

1 Introduction

In this lab, we will work on Ubuntu, which we established as a "victim" in previous labs. We will take actions such as experiencing different types of users, working on services such as Apache, OpenSSH, and Vsftpd, learning about system logs and structure, and securing the system. The operating system I am working on features:

- VMware WorkStation 15 Player
- Ubuntu 19.10 Desktop
- Number of processor cores: 2
- Hard Disk Capacity: 10 GB
- Memory: 2048 MB

2 Assignments

1) Be familiar with user and root (sudo) functions.
2) Install following services (apt-get install or ubuntu software centre):
    a. http-server -> Apache or other
    b. ssh-server -> Openssh or other
    c. ftp-server -> vsftpd or other

(You can try other network services like dns. But do not install “dhcp” which will introduce problems in our network)

3) How can you test if these services are working as they should?
4) Learn to make these services more secure by applying some common configurations like:
    a. Openssh configuration - SSH/OpenSSH/Configuring
    b. For others - google it and write your findings
5) Learn to look at system logs. Where are the system logs and these service logs (http, ssh, ftp)?
6) Where are the critical system files? How can we keep track if these system files are changed or not?
7) How can we learn who last connected to our system with ssh or telnet?
8) How can we prevent the user being root if they are connected by ssh?
9) Write and implement the commands to look for:
    a. unusual accounts
    b. unusual log entries
    c. sluggish system performance
    d. Excessive memory use
    e. Decrease in Disk space
    f. Unusual process and services
    g. Unusual files
    h. Unusual network usage
    i. Unusual scheduled tasks
10) Write a script which implements those

2.1 Assignment 1 (“Be familiar with user and root (sudo) functions”)

On a Linux operating system, there are three basic types of user accounts: root, regular and service. This assignment is about two of them; root and regular. While the Linux is installed, the root account that by default has access to all commands and files is automatically created. It is also known as super user. However, regular users have more limited access. They just have the necessary privileges to perform standard tasks like to store files in their own home directories.

2.1.1 Be familiar with the user functions

I am going to run very basic commands. Then, I am going to try to run the commands cannot run without being super user.

Commands that I used:

mkdir: allows you to create a new directory

touch: allows you to create a new document

ls: lists all files and folders in the current working directory

rm: removes the specified file

cd ..: takes you up one directory level

```
beyza@beyza-virtual-machine:~$ mkdir myDir
beyza@beyza-virtual-machine:~$ touch myDoc
beyza@beyza-virtual-machine:~$ ls
Desktop    Downloads  myDir  Pictures  report2    Videos
Documents  Music      myDoc  Public    Templates
beyza@beyza-virtual-machine:~$ rm report2
beyza@beyza-virtual-machine:~$ ls
Desktop    Downloads  myDir  Pictures  Templates
Documents  Music      myDoc  Public    Videos
beyza@beyza-virtual-machine:~$ cd ..
beyza@beyza-virtual-machine:/home$ mkdir newDir
mkdir: cannot create directory ‘newDir’: Permission denied
```

2.1.2 Be familiar with the root functions

First, I am going to be the super user with ‘su’ command by using my password. In other assignments, I am going to use commands with 'su'. Therefore, I am going to run just one harmless command with 'su' and try to run 'mkdir' command that could not run at home directory in the previous step.

New commands that I used:

su: allows you to run commands with the privileges of another user, by default the root user

apt-get update: to download package information from all configured sources

```
beyza@beyza-virtual-machine:~$ su
Password:
root@beyza-virtual-machine:/home/beyza# apt-get update
Hit:1 http://tr.archive.ubuntu.com/ubuntu eoan InRelease
Hit:2 http://tr.archive.ubuntu.com/ubuntu eoan-updates InRelease
Hit:3 http://tr.archive.ubuntu.com/ubuntu eoan-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu eoan-security InRelease [97,5 kB]
Fetched 97,5 kB in 1s (129 kB/s)
Reading package lists... Done
root@beyza-virtual-machine:/home/beyza# cd ..
root@beyza-virtual-machine:/home# mkdir newDir
root@beyza-virtual-machine:/home# ls
beyza  newDir
```

2.2 Assignment 2 (“Install following services:”)

- http-server -> Apache or other
- ssh-server -> Openssh or other
- ftp-server -> vsftpd or other

2.2.1 Install http-server -> Apache

Apache that is an open-source and free web server software turns a computer into an HTTP server. I am going to use a site as a guide to install Apache: “How to Install Apache on Ubuntu 18.04” from https://phoenixnap.com .

Step 1: Install Apache

I refreshed my local software package database to make sure I access the latest versions. Then, I installed the Apache package.

```
beyza@beyza-virtual-machine:~$ sudo apt-get update
Hit:1 http://tr.archive.ubuntu.com/ubuntu eoan InRelease
Hit:2 http://tr.archive.ubuntu.com/ubuntu eoan-updates InRelease
Hit:3 http://tr.archive.ubuntu.com/ubuntu eoan-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu eoan-security InRelease
Reading package lists... Done
beyza@beyza-virtual-machine:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
…
Rules updated for profile 'Apache'
Rules updated for profile 'OpenSSH'
Firewall reloaded
```

Step 2: Configure The Firewall

I should have displayed available app profiles on UFW and allowed normal web traffic on port 80. But I had did that at my Ceng3543 Assignment1. So, it did not add this as a new rule.

```
beyza@beyza-virtual-machine:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
```

```
beyza@beyza-virtual-machine:~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
beyza@beyza-virtual-machine:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

2.2.2 Install ssh-server -> Openssh

Short for Open Secure Shell, OpenSSH that is a free suite of tools help secure your network connections. OpenSSH encrypts all traffic to effectively eliminate eavesdropping, connection hijacking and other network-level attacks. I am going to use a site as a guide to install Openssh: “How to Enable Secure Shell (SSH) Service in Ubuntu 19.10” from https://tipsonubuntu.com .

The OpenSSH client package is installed by default in Ubuntu.To access to the current machine via ssh protocol, we can install the server package. I had just run "sudo apt-get update" command, so I didn't run it again. Once installed, SSH starts automatically in background.

```
beyza@beyza-virtual-machine:/$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass
The following NEW packages will be installed:
  openssh-server
0 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.
…
Rules updated for profile 'OpenSSH'
Firewall reloaded
```

2.2.3 Install FTP-server -> vsftpd

Vsftpd also known as a very secure FTP daemon is an FTP server for Unix-like systems. FTP is most widely used standard network protocol used for uploading/downloading files between two computers over a network. By default, FTP is insecure because it transmits data together with user credentials without encryption. I am going to use a site as a guide to install Openssh:

"Install and Configure VSFTPD server on Ubuntu 18.04 LTS" from https://www.howtoforge.com/tutorial/ubuntu-vsftpd/ .

Step 1: Install Vsftpd

```
beyza@beyza-virtual-machine:/$ sudo apt-get install vsftpd
[sudo] password for beyza:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.
Need to get 115 kB of archives.
…
Processing triggers for systemd (242-7ubuntu3.7) ...
```

- I started Vsftpd service and enable it to start on boot time.

```
beyza@beyza-virtual-machine:/$ sudo systemctl start vsftpd
beyza@beyza-virtual-machine:/$ sudo systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
```

Step 2: Create Directory Structure for FTP

- I created a user for FTP access.

```
beyza@beyza-virtual-machine:~$ sudo adduser vsftp
Adding user `vsftp' ...
Adding new group `vsftp' (1001) ...
Adding new user `vsftp' (1001) with group `vsftp' ...
Creating home directory `/home/vsftp' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for vsftp
Enter the new value, or press ENTER for the default
      Full Name []: beyza
      Room Number []: 1
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] Y
```

- I created ftp directory and set the ownership.

```
beyza@beyza-virtual-machine:~$ sudo mkdir /home/vsftp/ftp
beyza@beyza-virtual-machine:~$ sudo chown nobody:nogroup /home/vsftp/ftp
beyza@beyza-virtual-machine:~$ sudo chmod a-w /home/vsftp/ftp
```

- I created a directory where files can be uploaded and give the ownership to vsftp user.

```
beyza@beyza-virtual-machine:~$ sudo mkdir /home/vsftp/ftp/test
beyza@beyza-virtual-machine:~$ sudo chown vsftp:vsftp /home/vsftp/ftp/test
```

Step 3: Configure Vsftpd

- I created a backup of original config file before performing some configurations to setup FTP server.

```
beyza@beyza-virtual-machine:~$ sudo cp /etc/vsftpd.conf
/etc/vsftpd.conf.bak
[sudo] password for beyza:
```

- I opened the vsftpd.conf file with command "sudo nano /etc/vsftpd.conf" and deleted these lines' signs that make lines as a comment: *"write_enable=YES", "local_umask=022", "chroot_local_user=YES"*

- After I added the following lines at the end of the document, I saved it:

  "pasv_enable=Yes", "pasv_min_port=10000", "pasv_max_port=11000" , "user_sub_token=$USER", "local_root=/home/$USER/ftp", "userlist_enable=YES", "userlist_file=/etc/vsftpd.userlist", "userlist_deny=NO"

- I opened the file with "sudo nano /etc/vsftpd.userlist" command and added "*vsftp*" line to allow FTP.

- I restarted Vsftpd service to apply these changes.

```
beyza@beyza-virtual-machine:~$ sudo systemctl restart vsftpd
[sudo] password for beyza:
```

2.3 Assignment 3 (“How can you test if these services are working as they should?”)

2.3.1 Apache

- I opened “http://localhost/” website on a web browser to verify. It works!

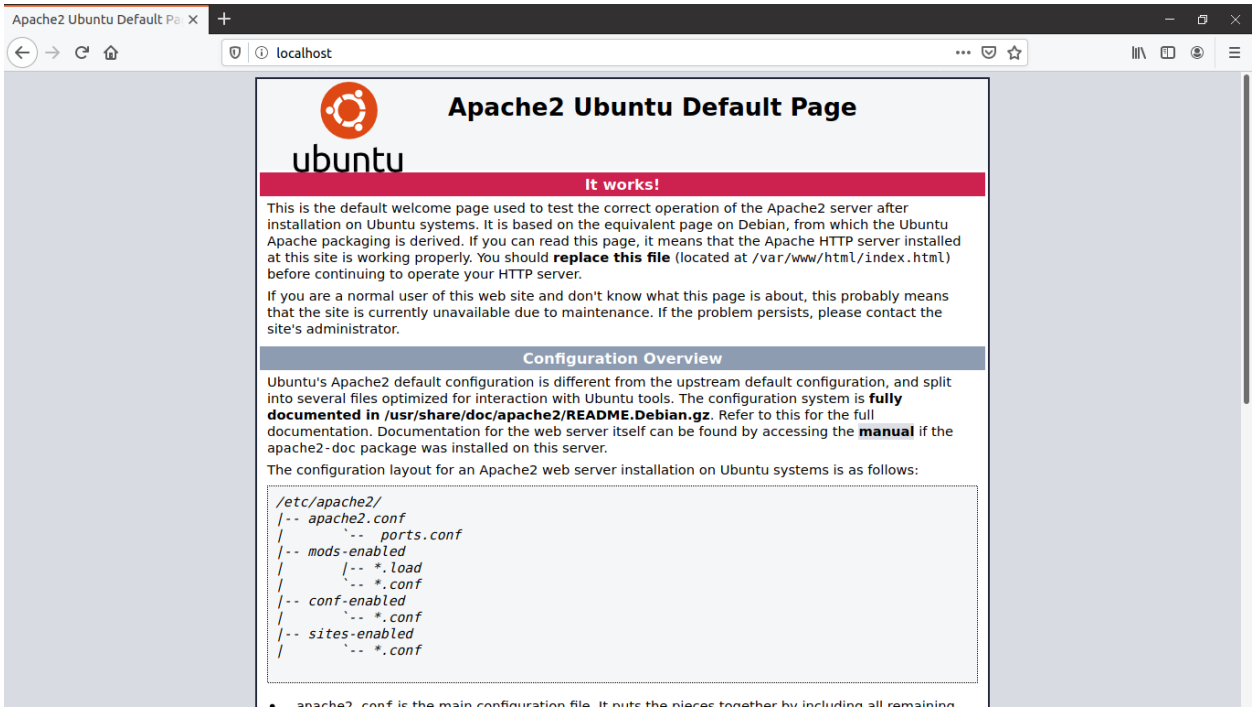

Figure 1 Apache2 Ubuntu Default Page

NOTE: It can also be verifyed with ip address instead of “localhost” in “https://localhost”.

- I also checked my Apache2 connection with the terminal, it works.

```
beyza@beyza-virtual-machine:~$ sudo systemctl check apache2
active
```

2.3.2 OpenSSH

I checked the ssh daemon status on the terminal. It works.

```
beyza@beyza-virtual-machine:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
   Active: active (running) since Sun 2020-05-10 03:42:12 +03; 1h 54min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
```

```
 Main PID: 956 (sshd)
    Tasks: 1 (limit: 2286)
   Memory: 2.1M
   CGroup: /system.slice/ssh.service
           └─956 /usr/sbin/sshd -D

May 10 03:42:10 beyza-virtual-machine systemd[1]: Starting OpenBSD Secure
Shell server...
May 10 03:42:12 beyza-virtual-machine sshd[956]: Server listening on
0.0.0.0 port 22.
May 10 03:42:12 beyza-virtual-machine sshd[956]: Server listening on ::
port 22.
May 10 03:42:12 beyza-virtual-machine systemd[1]: Started OpenBSD Secure
Shell server.
```

2.3.3 Vsftpd

- I opened "ftp://localhost/" website on a web browser to verify. It asked for username and password. But it returned nothing.

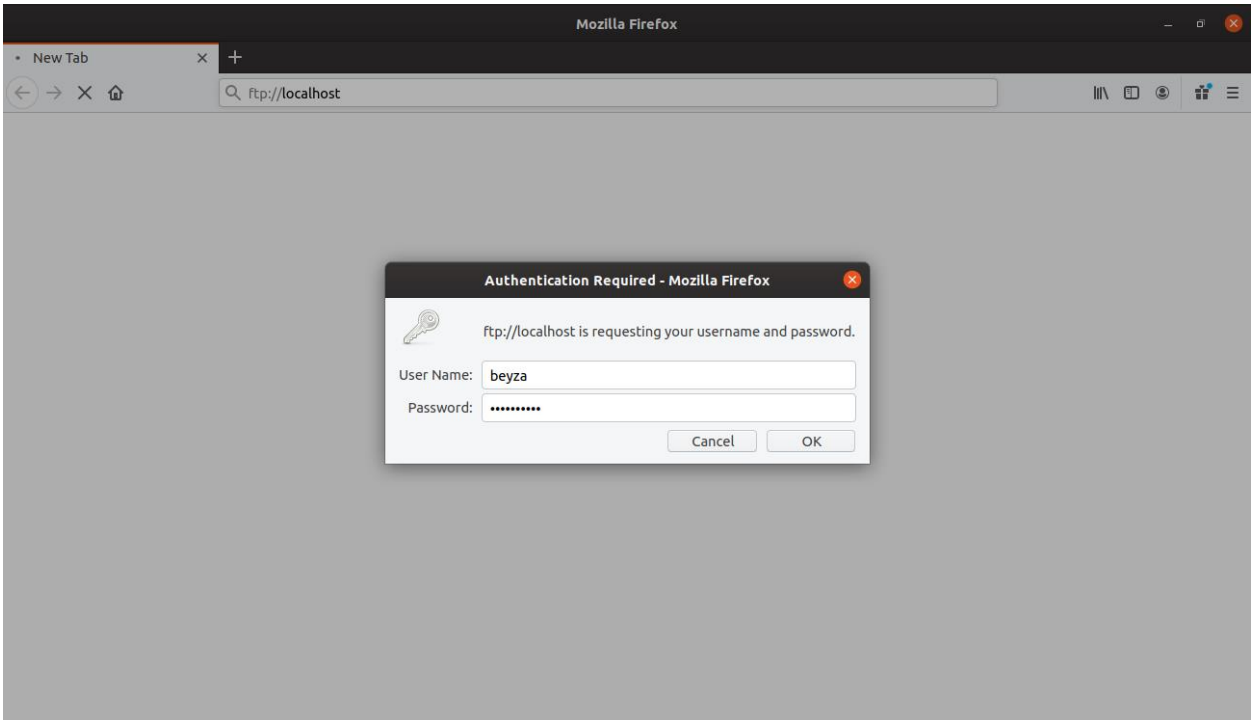

Figure 2 ftp://localhost/

- I also added "beyza" in "/etc/vsftpd.userlist" file, but nothing changed.
- UFW can block FTP traffic by default. So, I opened Ports 20 and 21 for FTP traffic.

```
beyza@beyza-virtual-machine:~$ sudo ufw allow 20/tcp
Rule added
Rule added (v6)
beyza@beyza-virtual-machine:~$ sudo ufw allow 21/tcp
Rule added
Rule added (v6)
beyza@beyza-virtual-machine:~$ sudo ufw reload
Firewall reloaded
beyza@beyza-virtual-machine:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
20/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
20/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
```

- Then I realized that I should write "*vsftp*" as username on authentication box, because of "*sudo adduser vsftp*" command I did. "*beyza*" is just full name. It works.

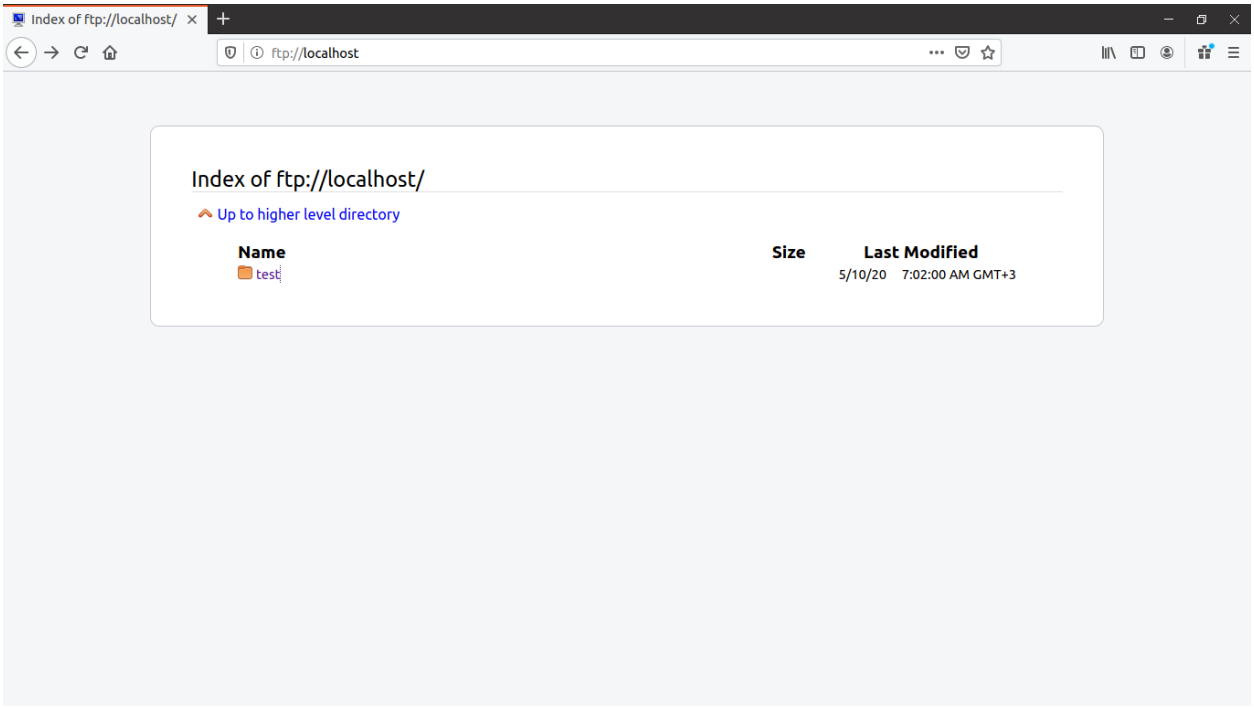

Figure 3 ftp://localhost/ (2)

2.4 Assignment 4 (“Learn to make the services more secure by applying some common configurations”)

2.4.1 Apache

For providing Apache security, I am going to follow “*13 Apache Web Server Security and Hardening Tips*” article’s advices from https://www.tecmint.com/.

A. Hide Apache Version and OS Identity from Errors

The article is saying that Apache displays the version of your Apache web server installed on your server with the Operating system name of your server in Errors by default.

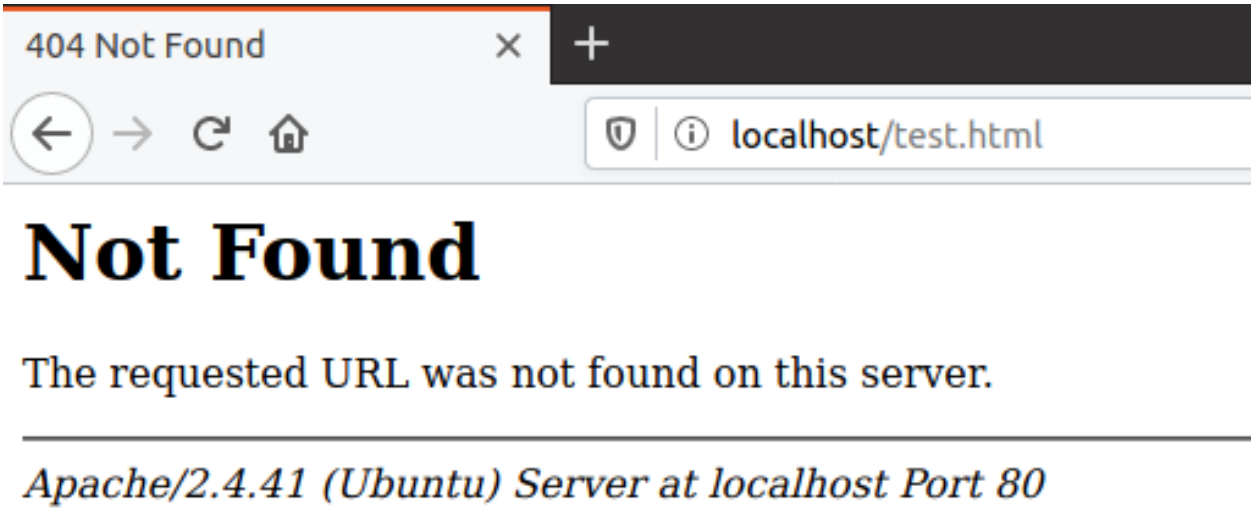

Figure 4 Apache Error Message with Informations

After I learn that, I tried to find a page does not exist for getting error and I achieved; my informations that must be private are seen in above picture. This can cause security problems. I am going to make little changes in Apache main configuration file to prevent the security problems.

The article says that “ServerSignature“ is on by default in configuration file and I should change it to off and to the second line add “ServerTokens Prod” that tells Apache to return only Apache as product in the server response header on the every page request. However, in my configure file, there was no line like “ServerSignature“. I added lines "***ServerSignature Off***" and “***ServerTokens Prod***” to the end of the file and restarted the service.

```
beyza@beyza-virtual-machine:~$ sudo nano /etc/apache2/apache2.conf
beyza@beyza-virtual-machine:~$ sudo service apache2 restart
```

As you can see in the picture below, my information does not appear on the error page.

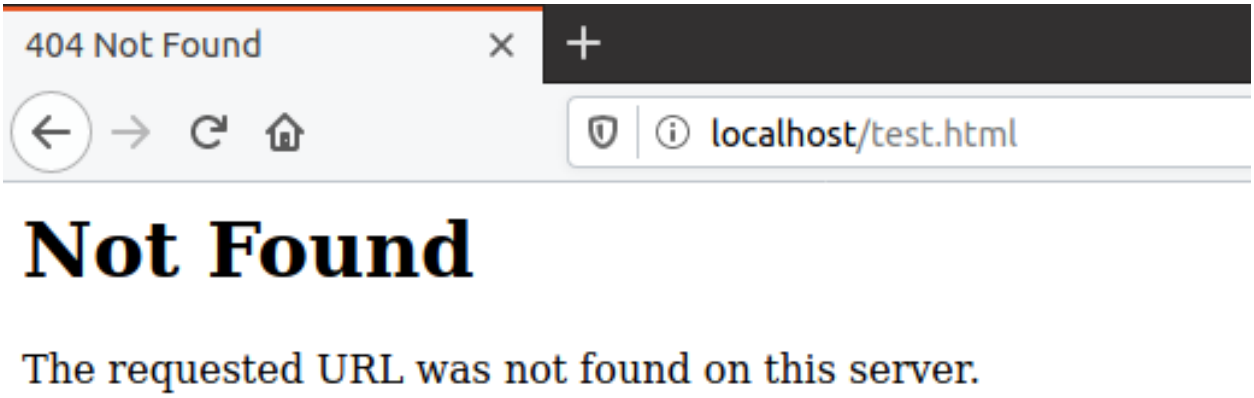

Figure 5 Apache Error Page

B. Disable Directory Listing

By default Apache list all the content of Document root directory in the absence of index file. It should be turned off directory listing by using Options directive in configuration file for a specific directory. I made an entry in apache2.conf file:

```
<Directory /var/www/html>
    Options -Indexes
</Directory>
```

```
beyza@beyza-virtual-machine:~$ sudo nano /etc/apache2/apache2.conf
[sudo] password for beyza:
```

The below pictures are from the article because I do not have a “Index of /” page.

Before configuration:

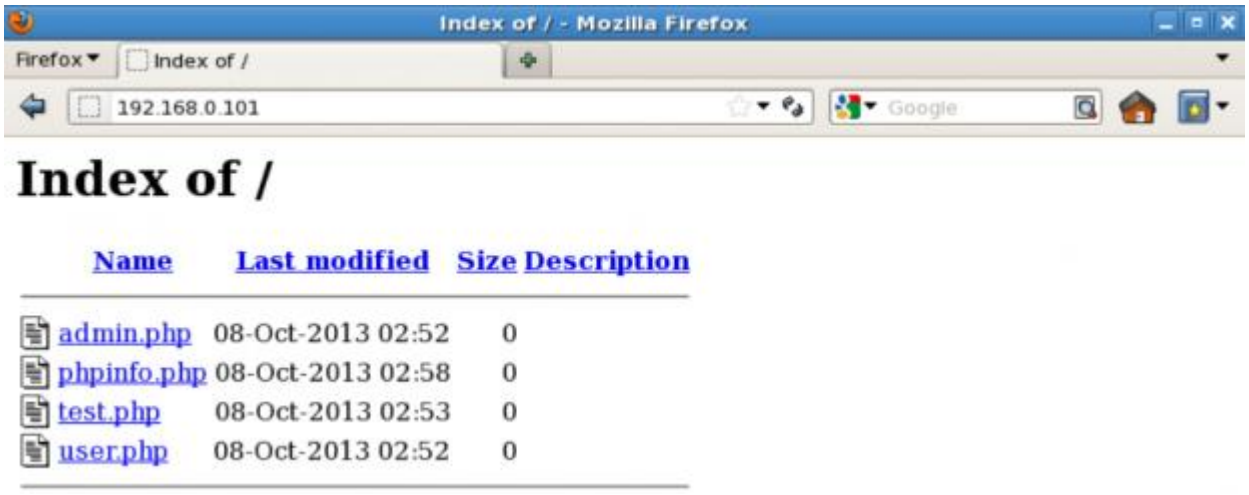

Figure 6 Apache Index of / Page

After configuratin:

Figure 7 Apache Forbidden Page

C. Keep Updating Apache Regularly

There may be security gaps that are still not noticed in the systems. These vulnerabilities can be found by users or developer teams. Therefore, the developer team is trying to close the gap as much as possible by bringing new updates regularly. So getting updates regularly makes our system safer.

Apache version can be controlled by "apache2 -v" command and be updated by "apt-get install apache2" command.

```
beyza@beyza-virtual-machine:~$ apache2 -v
Server version: Apache/2.4.41 (Ubuntu)
Server built:   2019-08-14T14:36:32
beyza@beyza-virtual-machine:~$ sudo apt-get install apache2
[sudo] password for beyza:
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.41-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
```

D. Use “mod_security” and “mod_evasive” Modules to Secure Apache

These two modules “mod_security” and “mod_evasive” are very popular modules of Apache in terms of security.

mod_security:

Where mod_security works as a firewall for our web applications and allows us to monitor traffic on a real time basis. It also helps us to protect our websites or web server from brute force attacks.

```
beyza@beyza-virtual-machine:/home/beyza$ sudo apt-get install
libapache2-mod-security2
Reading package lists... Done
Building dependency tree
Reading state information... Done
libapache2-mod-security2 is already the newest version (2.9.3-1).
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
beyza@beyza-virtual-machine:/home/beyza$ apache2 -t -D DUMP_MODULES
Loaded Modules:
 core_module (static)
 so_module (static)
…
 mime_module (shared)
 mpm_event_module (shared)
 negotiation_module (shared)
 reqtimeout_module (shared)
 security2_module (shared)
 setenvif_module (shared)
 status_module (shared)
 unique_id_module (shared)
```

- After installing the module successfully, I got an error when I wanted to list my modules. Since the virtual machine was frozen during the solution phase and I had to restart my computer, there is no output about that. The error message contained the sentence "Ubuntu Apache2 DefaultRuntimeDir must be a valid directory, absolute or relative to ServerRoot". When I looked at the file "/etc /apache2/envvars", there was the line " APACHE_RUN_DIR=/var/run/apache2$SUFFIX". I thought it was wrong because "SUFFIX" was empty and I changed it to "Beyza". This time it also failed while running Apache2. I changed the name of the "apache2" folder to "apache2Beyza" in the "/var/log" folder. Apache worked but when I wanted to list the modules again, I got the error message: "[Sun May 10 21: 10: 39.550845 2020] [core: warn] [pid 2493] AH00111: Config variable $ {APACHE_LOCK_DIR} is not defined
  [Sun May 10 21: 10: 39.550914 2020] [core: warn] [pid 2493] AH00111: Config variable $ {APACHE_RUN_DIR} is not defined
  apache2: Syntax error on line 80 of /etc/apache2/apache2.conf: DefaultRuntimeDir must be a valid directory, absolute or relative to ServerRoot"

  After running the command "source /etc/apache2/envvars", it bringed me to the main folder. When I run the command "apache2 -S", it returned with an error message:

  "AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message."

  I removed the comment sign where before "ServerRoot '/etc/apache2'" line in the "/etc/apache2/apache2.conf" file and added the text "ServerName '127.0.1.1'" to the line below it. When I run the command "apache2 -S" again, the error message did not appear and I successfully displayed my modules where I ran the command "apache2 -t -D DUMP_MODULES".

2.4.2 OpenSSH

For providing OpenSSH security, I am going to follow "*SSH/OpenSSH/Configuring*" article's advices from https://help.ubuntu.com.

Step 1: Configuration

I made a backup of the sshd_config file by copying it to my home directory with command:

"*sudo cp /etc/ssh/sshd_config / etc/ssh/sshd_config.factory-defaults*"

I made changes on it.

A. Disable Password Authentication

remove	add
#PasswordAuthentication yes	PasswordAuthentication no

Since most users use weak passwords on SSH servers, hackers can achieve their goals with a bit of hassle. The article said that "The recommended solution is to use SSH keys instead of passwords. To be as hard to guess as a normal SSH key, a password would have to contain 634 random letters and numbers. If you'll always be able to log in to your computer with an SSH key, you should disable password authentication altogether.".

B. Disable Forwarding

remove	add
AllowTcpForwarding yes	AllowTcpForwarding no
X11Forwarding yes	X11Forwarding no

To tunnel network connections and specific graphical applications through an SSH session can be useful but is not secure. They also give more options to an attacker who has already guessed your password. Disabling these options gives you a little security.

C. Rate-limit the connections

It's possible to limit the rate at which one IP address can establish new SSH connections by configuring the ufw. If an IP address is tries to connect more than 10 times in 30 seconds, all the following attempts will fail since the connections will be DROPped.

- I added a rule to the firewall by command:

  sudo ufw limit ssh

- I also added this line in my sshd_config file:

  "MaxStartups 2:30:10"

D. **Log More Information**

remove	add
LogLevel INFO	LogLevel VERBOSE

The article said that "By default, the OpenSSH server logs to the AUTH facility of syslog, at the INFO level. If you want to record more information - such as failed login attempts - you should increase the logging level to VERBOSE."

E. Display a Banner

remove	add
#Banner none	Banner /etc/issue

A warning message can be used to scare attackers. It doesn't provide security systematically, but it can work against some.

```
beyza@beyza-virtual-machine:/home/beyza$ sudo cp /etc/ssh/sshd_config
/etc/ssh/sshd_config.factory-defaults
[sudo] password for beyza:
beyza@beyza-virtual-machine:/home/beyza$ sudo nano /etc/ssh/sshd_config
beyza@beyza-virtual-machine:/home/beyza$ sudo ufw limit ssh
Rule added
Rule added (v6)
beyza@beyza-virtual-machine:/home/beyza$ sudo nano /etc/ssh/sshd_config
beyza@beyza-virtual-machine:/home/beyza$ sudo nano /etc/issue
```

Step 2: Troubleshooting

Once I have finished editing `sshd_config`, I made sure to save my changes before restarting my SSH daemon.

- I checked that my SSH daemon is running:

  ps -A | grep sshd
  ssh -v localhost

```
beyza@beyza-virtual-machine:/home/beyza$ ps -A | grep sshd
   893 ?        00:00:00 sshd
beyza@beyza-virtual-machine:/home/beyza$ ssh -v localhost
OpenSSH_8.0p1 Ubuntu-6build1, OpenSSL 1.1.1c  28 May 2019
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug1: Connecting to localhost [127.0.0.1] port 22.
…
debug1: kex: algorithm: curve25519-sha256
debug1: kex: host key algorithm: ecdsa-sha2-nistp256
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key:
SHA256:
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
debug1: rekey out after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
…
Welcome to Ubuntu 19.10 (GNU/Linux 5.3.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


10 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
beyza@beyza-virtual-machine:~$ sudo systemctl restart ssh
```

2.4.3 Vsftpd

For providing OpenSSH security, I am going to follow "*How To Secure vsFTPd With SSL/TLS*" article's advices from https://www.vultr.com/.

I removed "chroot_local_user=YES" line's comment sign.

A. Generate a Self Signed Certificate

The article says that a self signed certificate is typically used in a public key agreement protocol, you will now use openssl to generate a public key and a corresponding private key.

- I made a directory to store these two key files, preferably in a safe location normal users can not access.

  sudo mkdir -p /etc/vsftpd/ssl

- To the actual generation of the certificate, I stored both the keys in the same file (/etc/vsftpd/ssl/vsftpd.pem):

sudo openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout /etc/vsftpd/ssl/vsftpd.pem -out /etc/vsftpd/ssl/vsftpd.pem

This certificate will be valid for 365 days, it will use the RSA key agreement protocol with a key length of 4096 bits, and the file containing both the keys will be stored in the new directory I just created.

- To start using the new certificate and thus provide encryption, I added new lines to the configuration file.
  "sudo nano /etc/vsftpd.conf"

  - the paths to the new certificate and key files:
    rsa_cert_file=/etc/vsftpd/ssl/vsftpd.pem
    rsa_private_key_file=/etc/vsftpd/ssl/vsftpd.pem
  - *ssl_enable=YES*
  - to block anonymous users from using SSL:
    allow_anon_ssl=NO
  - to specify when to use SSL/TLS, this will enable encryption both for data transfer and login credentials:
    force_local_data_ssl=YES
    force_local_logins_ssl=YES
  - To specify what versions and protocols to be used. TLS is generally more secure than SSL and thus I may allow TLS and at the same time block older versions of SSL:
    ssl_tlsv1=YES
    ssl_sslv2=NO
    ssl_sslv3=NO
  - The article says that "Require SSL reuse and the usage of high ciphers will also help improve the security."
    require_ssl_reuse=YES
    ssl_ciphers=HIGH

```
beyza@beyza-virtual-machine:~$ sudo mkdir -p /etc/vsftpd/ssl
beyza@beyza-virtual-machine:~$ sudo openssl req -x509 -nodes -days 365 -
newkey rsa:4096 -keyout /etc/vsftpd/ssl/vsftpd.pem -out
/etc/vsftpd/ssl/vsftpd.pem
Generating a RSA private key
.....................................................................
.....................................................................
....++++
.....................................................................
.....................................................................
.....................++++
writing new private key to '/etc/vsftpd/ssl/vsftpd.pem'
-----
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TR
State or Province Name (full name) [Some-State]:Zonguldak
Locality Name (eg, city) []:Caycuma
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:beyzakurt1998@hotmail.com
beyza@beyza-virtual-machine:~$ sudo nano /etc/vsftpd.conf
beyza@beyza-virtual-machine:~$ /etc/init.d/vsftpd restart
[....] Restarting vsftpd (via systemctl): vsftpd.service====
AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'vsftpd.service'.
Authenticating as: beyza,,, (beyza)
Password:
==== AUTHENTICATION COMPLETE ===
. ok
```

2.5 Assignment 5 (“Learn to look at system logs. Where are the system logs and these service logs (http, ssh, ftp)?”)

Ubuntu maintains logs in different folders according to the kind of information they give. We can categorize them like "system", "application" and “non-human-readable” in general.

2.5.1 System logs

These logs are about the system things and contain information about authorizations, system daemons and system messages.

A. Authorization log

Location: /var/log/auth.log

Keeps track of authorization systems, such as password prompts, the sudo command and remote logins.

B. Daemon Log

Location: /var/log/daemon.log

Daemons are programs that run in the background, usually without user interaction. For example, display server, SSH sessions, printing services, bluetooth, and more.

C. Debug Log

Location: /var/log/debug

Provides debugging information from the Ubuntu system and applications.

D. Kernel Log

Location: /var/log/kern.log

Logs from the Linux kernel.

E. System Log

Location: /var/log/syslog

Contains more information about your system. If there is nothing in the other logs, it's probably here.

2.5.2 Application Logs

A. Apache

Location: /var/log/apache2/ (subdirectory)

Apache creates several log files in the /var/log/apache2/ subdirectory. The access.log file records all requests made to the server to access files. error.log records all errors thrown by the server.

B. Vsftpd

Location: /var/log/vsftpd.log

Vsftpd creates a log file. You can see actions like that:

Sun May 10 06:06:56 2020 [pid 6891] CONNECT: Client "::ffff:127.0.0.1"

Sun May 10 06:07:03 2020 [pid 6890] [vsftp] OK LOGIN: Client "::ffff:127.0.0.1"

2.6 Assignment 6 ("Where are the critical system files? How can we keep track if these system files are changed or not?")

First, I should give information about Linux filesystem hierarchy for understanding why system files are critical.

A. Critical System Files

Anyone that use computer can understand easily that they cannot make changes on all files. Ubuntu organizes files in a hierarchical tree, where relationships are thought of in teams of children and parent. Directories can contain other directories as well as regular files, which are the "leaves" of the tree. As mentioned at assignment 1, there are diffirent user types like

superuser or regular. While the superuser can make changes to all files, the regular user has more limited access than them and thus hierarchy gains meaning. I want to share a picture that can be seen in below; Linux directory structure. As mentioned, there are a root directory of the entire filesystem, main directories and standart directories. I am going to explain some of them that contain important files with some critical examples.

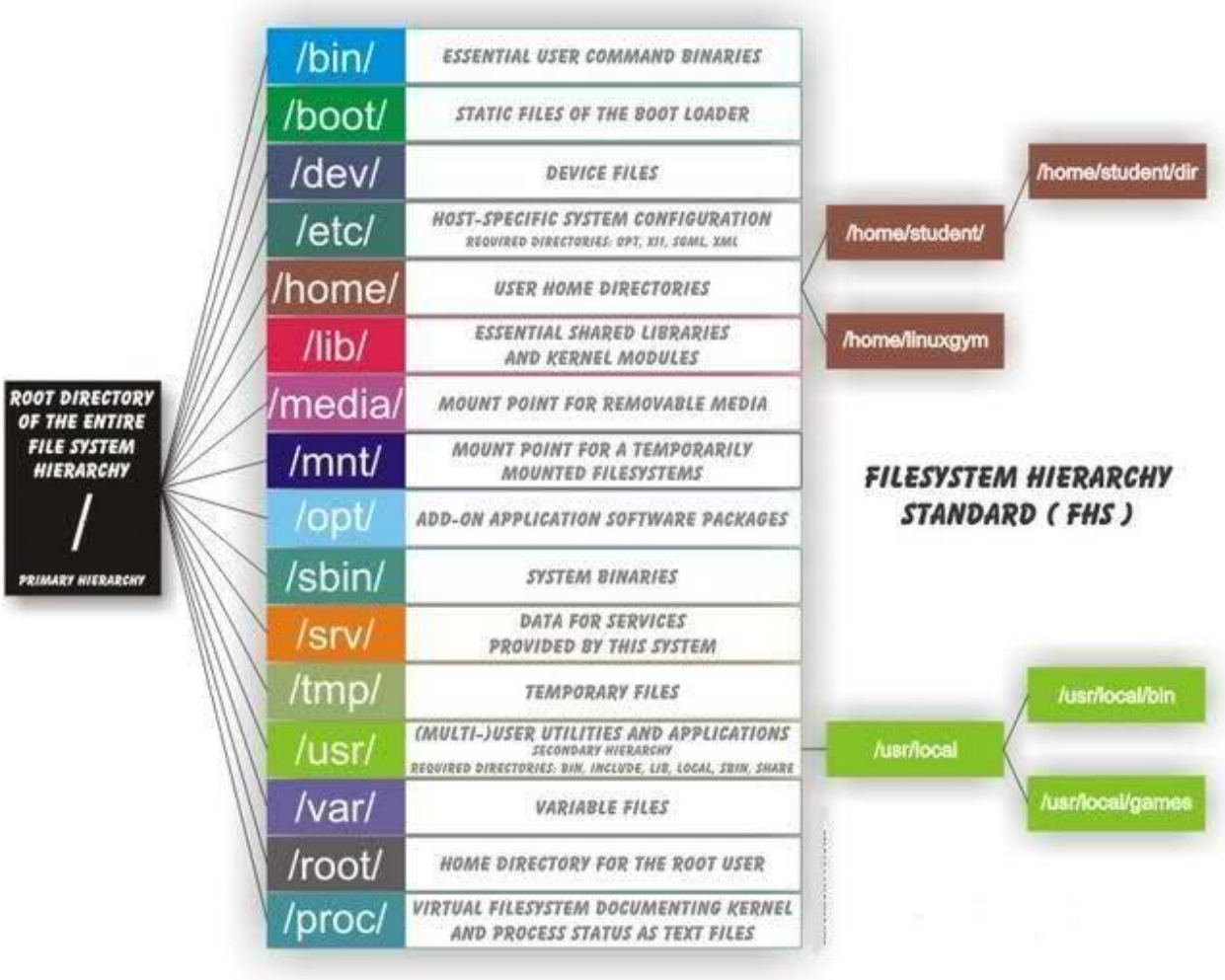

Figure 8 Linux directory structure

/boot/

Holds important files during boot-up process.

- /vmlinuz : The Linux Kernel file

/dev/

Contains device files for all the hardware devices on the machine e.g., cdrom, cpu.

- /hda : Device file for the first IDE HDD (Hard Disk Drive)
- /hdc : Device file for the IDE Cdrom, commonly

/etc/

Contains Application's configuration files, startup, shutdown, start, stop script for every individual program.

- /bashrc : Contains system defaults and aliases used by bash shell.
- /crontab : A shell script to run specified commands on a predefined time Interval.
- /fstab : Information of Disk Drive and their mount point.
- /group : Information of Security Group.
- /grub.conf : grub bootloader configuration file.
- /init.d : Service startup Script.
- /lilo.conf : lilo bootloader configuration file.
- /hosts : Information of Ip addresses and corresponding host names.
- /inittab : INIT process and their interaction at various run level.
- /modules.conf : Configuration files for system modules.
- /profile : Bash shell defaults

/usr/

Contains executable binaries, documentation, source code, libraries for second level program.

- /share : Shared directories of man files, info files, etc.
- /lib : Library files which are required during program compilation.
- /sbin : Commands for Super User, for System Administration.

/proc/

A virtual and pseudo file-system which contains information about running process with a particular Process-id aka pid.

- /filesystems : File-system Information being used currently.
- /ioports : Contains all the Input/Output addresses used by devices on the server.
- /modules : Currently using kernel module.
- /stat : Detailed Statistics of the current System.

- /swaps : Swap File Information.

/var/

Stands for variable. The contents of this file is expected to grow. This directory contains log, lock, spool, mail and temp files.

- /log/lastlog : log of last boot process.

B. Keep Track If These System Files are Changed or Not

There is one thing that shows all the changes in the specified directory after execution: inotify-tools. Its description is "monitoring filesystem events". Inotify can be used to monitor individual files, or to monitor directories. When a directory is monitored, inotify will return events for the directory itself, and for files inside the directory.

❖ I executed "inotifywait -m" with a system directory, "/var/log", and look up while "sudo apt-get update" command was running in other terminal.

```
beyza@beyza-virtual-machine:/var$ sudo inotifywait -m ./log
Setting up watches.
Watches established.
./log/ MODIFY vmware-vmsvc-root.log
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY vmware-vmsvc-root.log
./log/ MODIFY vmware-vmsvc-root.log
./log/ MODIFY auth.log
./log/ MODIFY auth.log
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY syslog
./log/ MODIFY auth.log
./log/ MODIFY vmware-vmsvc-root.log
^C
```

2.7 Assignment 7 (“How can we learn who last connected to our system with ssh or telnet?”)

As I said in the assignment 5, /var/log/auth.log keeps track of authorization systems, such as remote logins. We can examine this file or we can use "lastlog" command to view the last time each user on the system logged in.

2.8 Assignment 8 (“How can we prevent the user being root if they are connected by ssh?”)

/etc/ssh/sshd_config can be modify. "PermitRootLogin no" line to disable SSH logins for root and "DenyUsers root" line to deny root user access can be add.

2.9 Assignment 9 (“Write and implement the commands to look for:”)

a. Unusual accounts

```
beyza@beyza-virtual-machine:~$ function controlAccounts () {
> cd ~
> if [[ ! -d ~/AccountsControl ]]; then
>     echo creating a directory, “AccountsControl”, to the home…
>     mkdir ~/AccountsControl
>     cd ~/AccountsControl
>     echo You are in “AccountsControl”.
>     touch accOld.txt
> elif [[ ! -f ~/AccountsControl/accOld.txt ]]; then
>     cd ~/AccountsControl
>     echo You are in “AccountsControl”.
>     touch accOld.txt
> else
>     cd ~/AccountsControl
>     echo You are in “AccountsControl”.
> fi
> cat /etc/passwd > accLast.txt
> echo Accounts results were written to “accLast.txt”.
> [ -s accOld.txt ]
> echo $? > temp
> if [[ $(< temp) == "1" ]]; then
>     printf "There is no old record for account comparation.\n"
>     cat accLast.txt > accOld.txt
>     echo Accounts results were copied to “accOld.text” for future
comparisons.
> elif cmp -s accOld.txt accLast.txt; then
>     printf "There is no change in accounts.\n"
> else
>     printf 'There are new accounts!\n\n'
>     diff accOld.txt accLast.txt > results
```

```
>     echo Differences between old and last results are written to
“results”.
>     diff accOld.txt accLast.txt
>     cat accLast.txt > accOld.txt
>     echo Accounts results were copied to “accOld.txt” from "accLast.txt".
> fi
> rm temp
> }
beyza@beyza-virtual-machine:~$ controlAccounts
creating a directory, “AccountsControl”, to the home…
You are in “AccountsControl”.
Accounts results were written to “accLast.txt”.
There is no old record for account comparation.
Accounts results were copied to “accOld.text” for future comparisons.
beyza@beyza-virtual-machine:~/AccountsControl$ controlAccounts
You are in “AccountsControl”.
Accounts results were written to “accLast.txt”.
There is no change in accounts.
beyza@beyza-virtual-machine:~/AccountsControl$ ls
accLast.txt  accOld.txt
beyza@beyza-virtual-machine:~/AccountsControl$ cat accLast.txt
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
…
```

c. slugging system performance

```
beyza@beyza-virtual-machine:/$ function CPUBenchmark() {
> echo "sysbench is installing…"
> echo ""
> sudo apt install -y sysbench 2>/dev/null | sysbench cpu run
> }
beyza@beyza-virtual-machine:/$ CPUBenchmark
sysbench is installing…

Reading package lists... Done
Building dependency tree
Reading state information... Done
sysbench is already the newest version (1.0.17+ds-1).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.

------------------------------------------------------
sysbench 1.0.17 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
```

```
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1029.55

General statistics:
    total time:                          10.0009s
    total number of events:              10298

Latency (ms):
         min:                                    0.91
         avg:                                    0.97
         max:                                    2.62
         95th percentile:                        1.06
         sum:                                 9993.62

Threads fairness:
    events (avg/stddev):           10298.0000/0.00
    execution time (avg/stddev):   9.9936/0.00
```

d. Excessive memory use

```
beyza@beyza-virtual-machine:/$ function memoryBenchmark () {
> echo "sysbench is installing…"
> echo ""
> sudo apt install sysbench
> echo ""
> echo "------------------------------------------------------------"
> sysbench memory run
> }
@beyza-virtual-machine:/$ memoryBenchmark
sysbench is installing…

Reading package lists... Done
Building dependency tree
Reading state information... Done
sysbench is already the newest version (1.0.17+ds-1).
```

```
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.

--------------------------------------------------
sysbench 1.0.17 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 47843063 (4783519.90 per second)

46721.74 MiB transferred (4671.41 MiB/sec)


General statistics:
    total time:                          10.0000s
    total number of events:              47843063

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    2.85
         95th percentile:                        0.00
         sum:                                 4822.29

Threads fairness:
    events (avg/stddev):           47843063.0000/0.00
    execution time (avg/stddev):   4.8223/0.00
```

e. Decrease Disk Space

```
beyza@beyza-virtual-machine:~$ function controlDiskSpace () {
> cd ~
> if [[ ! -d ~/DiskSpaceControl ]]; then
>     echo creating a directory, “DiskSpaceControl”, to the home…
>     mkdir ~/DiskSpaceControl
>     cd ~/DiskSpaceControl
```

```
>     echo You are in “DiskSpaceControl”.
>     touch dSold.txt
> elif [[ ! -f ~/DiskSpaceControl/dSold.txt ]]; then
>     cd ~/DiskSpaceControl
>     echo You are in “DiskSpaceControl”.
>     touch dSold.txt
> else
>     cd ~/DiskSpaceControl
>     echo You are in “DiskSpaceControl”.
> fi
> df -h > dSlast.txt
> echo Disk space results were written to “dSlast.txt”.
> [ -s dSold.txt ]
> echo $? > temp
> if [[ $(< temp) == "1" ]]; then
>     printf "There is no old record for disk space comparation.\n"
>     cat dSlast.txt > dSold.txt
>     echo Disk space results were copied to “dSold.text” for future
comparisons.
> elif cmp -s dSold.txt dSlast.txt; then
>     printf "There is no change in disk space.\n"
> else
>     printf 'Disk space was changed!\n\n'
>     diff dSold.txt dSlast.txt > results
>     echo Differences between old and last results are written to
“results”.
>     diff dSold.txt dSlast.txt
>     cat dSlast.txt > dSold.txt
>     echo Disk space results were copied to “dSold.txt” from "dSlast.txt".
> fi
> rm temp
> }
beyza@beyza-virtual-machine:~$ controlDiskSpace
creating a directory, “DiskSpaceControl”, to the home…
You are in “DiskSpaceControl”.
Disk space results were written to “dSlast.txt”.
There is no old record for disk space comparation.
Disk space results were copied to “dSold.text” for future comparisons.
beyza@beyza-virtual-machine:~/DiskSpaceControl$ ls
dSlast.txt  dSold.txt
beyza@beyza-virtual-machine:~/DiskSpaceControl$ cat dSlast.txt
Filesystem      Size  Used Avail Use% Mounted on
udev            953M     0  953M   0% /dev
tmpfs           196M  1,5M  195M   1% /run
/dev/sda1       9,8G  6,6G  2,8G  71% /
tmpfs           980M   35M  946M   4% /dev/shm
tmpfs           5,0M  4,0K  5,0M   1% /run/lock
tmpfs           980M     0  980M   0% /sys/fs/cgroup
/dev/loop0       94M   94M     0 100% /snap/core/8935
/dev/loop3      2,5M  2,5M     0 100% /snap/gnome-calculator/748
```

```
/dev/loop1       55M   55M     0 100% /snap/core18/1754
/dev/loop2      161M  161M     0 100% /snap/gnome-3-28-1804/116
/dev/loop4       94M   94M     0 100% /snap/core/9066
/dev/loop5      2,5M  2,5M     0 100% /snap/gnome-calculator/730
/dev/loop8      150M  150M     0 100% /snap/gnome-3-28-1804/71
/dev/loop9      256M  256M     0 100% /snap/gnome-3-34-1804/33
/dev/loop7       55M   55M     0 100% /snap/core18/1705
/dev/loop6       63M   63M     0 100% /snap/gtk-common-themes/1506
/dev/loop13      15M   15M     0 100% /snap/gnome-characters/495
/dev/loop12     1,0M  1,0M     0 100% /snap/gnome-logs/93
/dev/loop15     243M  243M     0 100% /snap/gnome-3-34-1804/27
/dev/loop10      55M   55M     0 100% /snap/gtk-common-themes/1502
/dev/loop11     384K  384K     0 100% /snap/gnome-characters/539
/dev/loop14     1,0M  1,0M     0 100% /snap/gnome-logs/100
tmpfs           196M   32K  196M   1% /run/user/1000
/dev/sr0        2,3G  2,3G     0 100% /media/beyza/Ubuntu 19.10 amd64
beyza@beyza-virtual-machine:~/DiskSpaceControl$ cat dSold.txt
Filesystem      Size  Used Avail Use% Mounted on
udev            953M     0  953M   0% /dev
tmpfs           196M  1,5M  195M   1% /run
/dev/sda1       9,8G  6,6G  2,8G  71% /
tmpfs           980M   35M  946M   4% /dev/shm
tmpfs           5,0M  4,0K  5,0M   1% /run/lock
tmpfs           980M     0  980M   0% /sys/fs/cgroup
/dev/loop0       94M   94M     0 100% /snap/core/8935
/dev/loop3      2,5M  2,5M     0 100% /snap/gnome-calculator/748
/dev/loop1       55M   55M     0 100% /snap/core18/1754
/dev/loop2      161M  161M     0 100% /snap/gnome-3-28-1804/116
/dev/loop4       94M   94M     0 100% /snap/core/9066
/dev/loop5      2,5M  2,5M     0 100% /snap/gnome-calculator/730
/dev/loop8      150M  150M     0 100% /snap/gnome-3-28-1804/71
/dev/loop9      256M  256M     0 100% /snap/gnome-3-34-1804/33
/dev/loop7       55M   55M     0 100% /snap/core18/1705
/dev/loop6       63M   63M     0 100% /snap/gtk-common-themes/1506
/dev/loop13      15M   15M     0 100% /snap/gnome-characters/495
/dev/loop12     1,0M  1,0M     0 100% /snap/gnome-logs/93
/dev/loop15     243M  243M     0 100% /snap/gnome-3-34-1804/27
/dev/loop10      55M   55M     0 100% /snap/gtk-common-themes/1502
/dev/loop11     384K  384K     0 100% /snap/gnome-characters/539
/dev/loop14     1,0M  1,0M     0 100% /snap/gnome-logs/100
tmpfs           196M   32K  196M   1% /run/user/1000
/dev/sr0        2,3G  2,3G     0 100% /media/beyza/Ubuntu 19.10 amd64
beyza@beyza-virtual-machine:~/DiskSpaceControl$ test
You are in “DiskSpaceControl”.
Disk space results were written to “dSlast.txt”.
There is no change in disk space.
```

f. Unusual Process and Services

```
beyza@beyza-virtual-machine:~$ function controlProcess () {
> cd ~
> if [[ ! -d ~/ProcessControl ]]; then
>     echo creating a directory, “ProcessControl”, to the home…
>     mkdir ~/ProcessControl
>     cd ~/ProcessControl
>     echo You are in “ProcessControl”.
>     touch processOld.txt
> elif [[ ! -f ~/ProcessControl/processOld.txt ]]; then
>     cd ~/ProcessControl
>     echo You are in “ProcessControl”.
>     touch processOld.txt
> else
>     cd ~/ProcessControl
>     echo You are in “ProcessControl”.
> fi
> sudo ps aux > processLast.txt
> echo Process results were written to “processLast.txt”.
> [ -s processOld.txt ]
> echo $? > temp
> if [[ $(< temp) == "1" ]]; then
>     printf "There is no old record for process comparation.\n"
>     cat processLast.txt > processOld.txt
>     echo Process results were copied to “processOld.text” for future comparisons.
> elif cmp -s processOld.txt processLast.txt; then
>     printf "There is no change in processounts.\n"
> else
>     printf 'There are new process’!\n\n'
>     diff processOld.txt processLast.txt > results
>     echo Differences between old and last results are written to “results”.
>     diff processOld.txt processLast.txt
>     cat processLast.txt > processOld.txt
>     echo Process results were copied to “processOld.txt” from "processLast.txt".
> fi
> rm temp
> }
beyza@beyza-virtual-machine:~$ function controlServices () {
> cd ~
> if [[ ! -d ~/ServicesControl ]]; then
>     echo creating a directory, “ServicesControl”, to the home…
>     mkdir ~/ServicesControl
>     cd ~/ServicesControl
>     echo You are in “ServicesControl”.
>     touch serOld.txt
> elif [[ ! -f ~/ServicesControl/serOld.txt ]]; then
>     cd ~/ServicesControl
>     echo You are in “ServicesControl”.
>     touch serOld.txt
> else
>     cd ~/ServicesControl
```

```
>     echo You are in “ServicesControl”.
> fi
> sudo service  --status-all > serLast.txt
> echo Services results were written to “serLast.txt”.
> [ -s serOld.txt ]
> echo $? > temp
> if [[ $(< temp) == "1" ]]; then
>     printf "There is no old record for services comparation.\n"
>     cat serLast.txt > serOld.txt
>     echo Services results were copied to “serOld.text” for future comparisons.
> elif cmp -s serOld.txt serLast.txt; then
>     printf "There is no change in serounts.\n"
> else
>     printf 'There are new sevices!\n\n'
>     diff serOld.txt serLast.txt > results
>     echo Differences between old and last results are written to “results”.
>     diff serOld.txt serLast.txt
>     serLast.txt > serOld.txt
>     echo Services results were copied to “serOld.txt” from "serLast.txt".
> fi
> rm temp
> }
beyza@beyza-virtual-machine:~$ controlProcess
creating a directory, “ProcessControl”, to the home…
You are in “ProcessControl”.
Process results were written to “processLast.txt”.
There is no old record for process comparation.
Process results were copied to “processOld.text” for future comparisons.
beyza@beyza-virtual-machine:~/ProcessControl$ controlProcess
You are in “ProcessControl”.
Process results were written to “processLast.txt”.
There are new process’!

Differences between old and last results are written to “results”.
231c231
< beyza      1472  0.9  8.2 2660960 165692 ?      Rsl  09:31   0:28 /usr/bin/gnome-
shell
---
> beyza      1472  0.9  8.2 2660960 165692 ?      Ssl  09:31   0:28 /usr/bin/gnome-
shell
288,290c288,290
< root       9869  0.0  0.2  13980  4696 pts/0    S+   10:21   0:00 sudo ps aux
< beyza      9870  0.0  0.7 358824 15452 ?        Rsl  10:21   0:00
/usr/lib/tracker/tracker-store
< root       9871  0.0  0.1  14184  3628 pts/0    R+   10:21   0:00 ps aux
---
> beyza      9870  3.5  1.2 438376 24928 ?        Ssl  10:21   0:00
/usr/lib/tracker/tracker-store
> root       9881  0.0  0.2  13980  4700 pts/0    S+   10:21   0:00 sudo ps aux
> root       9882  0.0  0.1  14184  3680 pts/0    R+   10:21   0:00 ps aux
```

```
Process results were copied to “processOld.txt” from processLast.txt.
beyza@beyza-virtual-machine:~/ProcessControl$ controlServices
creating a directory, “ServicesControl”, to the home…
You are in “ServicesControl”.
Services results were written to “serLast.txt”.
There is no old record for services comparation.
Services results were copied to “serOld.text” for future comparisons.
beyza@beyza-virtual-machine:~/ServicesControl$ controlServices
You are in “ServicesControl”.
Services results were written to “serLast.txt”.
There is no change in services.
```

g. Unusual Files

```
beyza@beyza-virtual-machine:~$ function controlFiles () {
> cd ~
> if [[ ! -d ~/FilesControl ]]; then
>     echo creating a directory, “FilesControl”, to the home…
>     mkdir ~/FilesControl
>     cd ~/FilesControl
>     echo You are in “FilesControl”.
>     touch fileOld.txt
> elif [[ ! -f ~/FilesControl/fileOld.txt ]]; then
>     cd ~/FilesControl
>     echo You are in “FilesControl”.
>     touch fileOld.txt
> else
>     cd ~/FilesControl
>     echo You are in “FilesControl”.
> fi
> sudo apt install tree
> tree -a > fileLast.txt
> echo Files results were written to “fileLast.txt”.
> [ -s fileOld.txt ]
> echo $? > temp
> if [[ $(< temp) == "1" ]]; then
>     printf "There is no old record for files comparation.\n"
>     cat fileLast.txt > fileOld.txt
>     echo Files results were copied to “fileOld.text” for future
comparisons.
> elif cmp -s fileOld.txt fileLast.txt; then
>     printf "There is no change on files.\n"
> else
>     printf 'There are changes on files!\n\n'
>     diff fileOld.txt fileLast.txt > results
>     echo Differences between old and last results are written to
“results”.
>     diff fileOld.txt fileLast.txt
```

```
>     cat fileLast.txt > fileOld.txt
>     echo Files results were copied to “fileOld.txt” from "fileLast.txt".
> fi
> rm temp
> }
beyza@beyza-virtual-machine:~$ controlFiles
creating a directory, “FilesControl”, to the home…
You are in “FilesControl”.
Reading package lists... Done
Building dependency tree
Reading state information... Done
tree is already the newest version (1.8.0-1).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
Files results were written to “fileLast.txt”.
There is no old record for files comparation.
Files results were copied to “fileOld.text” for future comparisons.
beyza@beyza-virtual-machine:~/FilesControl$ controlFiles
You are in “FilesControl”.
Reading package lists... Done
Building dependency tree
Reading state information... Done
tree is already the newest version (1.8.0-1).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
Files results were written to “fileLast.txt”.
There is no change on files.
```

i. Unusual scheduled tasks

```
beyza@beyza-virtual-machine:~$ function controlTasks () {
> cd ~
> if [[ ! -d ~/TasksControl ]]; then
>     echo creating a directory, “TasksControl”, to the home…
>     mkdir ~/TasksControl
>     cd ~/TasksControl
>     echo You are in “TasksControl”.
>     touch taskOld.txt
> elif [[ ! -f ~/TasksControl/taskOld.txt ]]; then
>     cd ~/TasksControl
>     echo You are in “TasksControl”.
>     touch taskOld.txt
> else
>     cd ~/TasksControl
>     echo You are in “TasksControl”.
> fi
> crontab -l > taskLast.txt
> echo Tasks results were written to “taskLast.txt”.
> [ -s taskOld.txt ]
> echo $? > temp
```

```
> if [[ $(< temp) == "1" ]]; then
>     printf "There is no old record for Tasks comparation.\n"
>     cat taskLast.txt > taskOld.txt
>     echo Tasks results were copied to “taskOld.text” for future
comparisons.
> elif cmp -s taskOld.txt taskLast.txt; then
>     printf "There is no change on Tasks.\n"
> else
>     printf 'There are changes on Tasks!\n\n'
>     diff taskOld.txt taskLast.txt > results
>     echo Differences between old and last results are written to
“results”.
>     diff taskOld.txt taskLast.txt
>     cat taskLast.txt > taskOld.txt
>     echo Tasks results were copied to “taskOld.txt” from "taskLast.txt".
> fi
> rm temp
> }
beyza@beyza-virtual-machine:~$ controlTasks
creating a directory, “TasksControl”, to the home…
You are in “TasksControl”.
no crontab for beyza
Tasks results were written to “taskLast.txt”.
There is no old record for Tasks comparation.
Tasks results were copied to “taskOld.text” for future comparisons.
```

2.10 Assignment 10 (“Write a script which implements those”)

```
#!/bin/bash
clear

function controlAccounts () {
        cd ~
        if [[ ! -d ~/AccountsControl ]]; then
           echo creating a directory, “AccountsControl”, to the home…
           mkdir ~/AccountsControl
           cd ~/AccountsControl
           echo You are in “AccountsControl”.
           touch accOld.txt
        elif [[ ! -f ~/AccountsControl/accOld.txt ]]; then
           cd ~/AccountsControl
           echo You are in “AccountsControl”.
           touch accOld.txt
        else
           cd ~/AccountsControl
           echo You are in “AccountsControl”.
```

```
        fi

        cat /etc/passwd > accLast.txt
        echo Accounts results were written to “accLast.txt”.
        [ -s accOld.txt ]
        echo $? > temp

        if [[ $(< temp) == "1" ]]; then
           printf "There is no old record for account comparation.\n"
           cat accLast.txt > accOld.txt
           echo "Accounts results were copied to “accOld.text” for future comparisons."
        elif cmp -s accOld.txt accLast.txt; then
           echo There is no change in accounts.
        else
           printf 'There are new accounts!\n\n'
           diff accOld.txt accLast.txt > results
           echo "Differences between old and last results are written to “results”."
           diff accOld.txt accLast.txt
           cat accLast.txt > accOld.txt
           echo "Accounts results were copied to “accOld.txt” from 'accLast.txt'".
        fi
        rm temp
}

function CPUBenchmark() {
        echo "sysbench is installing..."
        echo ""
        sudo apt install -y sysbench
        sysbench cpu run
}

function memoryBenchmark () {
        echo ""
        sysbench memory run
}

function controlDiskSpace () {
        cd ~
        if [[ ! -d ~/DiskSpaceControl ]]; then
           echo creating a directory, “DiskSpaceControl”, to the home...
           mkdir ~/DiskSpaceControl
           cd ~/DiskSpaceControl
           echo "You are in “DiskSpaceControl”."
           touch dSold.txt
```

```
        elif [[ ! -f ~/DiskSpaceControl/dSold.txt ]]; then
            cd ~/DiskSpaceControl
            echo "You are in “DiskSpaceControl”".
            touch dSold.txt
        else
            cd ~/DiskSpaceControl
            echo "You are in “DiskSpaceControl”."
        fi

        df -h > dSlast.txt
        echo Disk space results were written to “dSlast.txt”.
        [ -s dSold.txt ]
        echo $? > temp

        if [[ $(< temp) == "1" ]]; then
            printf "There is no old record for disk space comparation.\n"
            cat dSlast.txt > dSold.txt
            echo "Disk space results were copied to “dSold.text” for future comparisons."
        elif cmp -s dSold.txt dSlast.txt; then
            printf "There is no change in disk space.\n"
        else
            printf "Disk space was changed!"
            diff dSold.txt dSlast.txt > results
            echo "Differences between old and last results are written to “results”."
            diff dSold.txt dSlast.txt
            cat dSlast.txt > dSold.txt
            echo "Disk space results were copied to “dSold.txt” from 'dSlast.txt'".
        fi
        rm temp
}

function controlProcess () {
        cd ~
        if [[ ! -d ~/ProcessControl ]]; then
            echo creating a directory, “ProcessControl”, to the home…
            mkdir ~/ProcessControl
            cd ~/ProcessControl
            echo "You are in “ProcessControl”."
            touch processOld.txt
        elif [[ ! -f ~/ProcessControl/processOld.txt ]]; then
            cd ~/ProcessControl
            echo "You are in “ProcessControl”."
            touch processOld.txt
        else
```

```
        cd ~/ProcessControl
        echo "You are in “ProcessControl”."
    fi
    sudo ps -aux > processLast.txt
    echo Process results were written to “processLast.txt”.
    [ -s processOld.txt ]
    echo $? > temp
    if [[ $(< temp) == "1" ]]; then
        printf "There is no old record for process comparation.\n"
        cat processLast.txt > processOld.txt
        echo "Process results were copied to “processOld.text” for future comparisons."
    elif cmp -s processOld.txt processLast.txt; then
        printf "There is no change in processounts.\n"
    else
        printf 'There are new process’!\n\n'
        diff processOld.txt processLast.txt > results
        echo "Differences between old and last results are written to “results”."
        diff processOld.txt processLast.txt
        cat processLast.txt > processOld.txt
        echo "Process results were copied to “processOld.txt” from 'processLast.txt'".
    fi
    rm temp
}

function controlServices () {
    cd ~
    if [[ ! -d ~/ServicesControl ]]; then
        echo creating a directory, “ServicesControl”, to the home…
        mkdir ~/ServicesControl
        cd ~/ServicesControl
        echo "You are in “ServicesControl”."
        touch serOld.txt
    elif [[ ! -f ~/ServicesControl/serOld.txt ]]; then
        cd ~/ServicesControl
        echo "You are in “ServicesControl”."
        touch serOld.txt
    else
        cd ~/ServicesControl
        echo You are in “ServicesControl”.
    fi

    sudo service --status -all > serLast.txt
    echo Services results were written to “serLast.txt”.
    [ -s serOld.txt ]
```

```
        echo $? > temp

        if [[ $(< temp) == "1" ]]; then
            printf "There is no old record for services comparation.\n"
            cat serLast.txt > serOld.txt
            echo "Services results were copied to “serOld.text” for future comparisons."
        elif cmp -s serOld.txt serLast.txt; then
            printf "There is no change in serounts.\n"
        else
            printf 'There are new sevices!\n\n'
            diff serOld.txt serLast.txt > results
            echo "Differences between old and last results are written to “results”."
            diff serOld.txt serLast.txt
            serLast.txt > serOld.txt
            echo "Services results were copied to “serOld.txt” from 'serLast.txt'".
        fi
        rm temp
}

function controlFiles () {
        cd ~
        if [[ ! -d ~/FilesControl ]]; then
            echo creating a directory, “FilesControl”, to the home…
            mkdir ~/FilesControl
            cd ~/FilesControl
            echo "You are in “FilesControl”."
            touch fileOld.txt
        elif [[ ! -f ~/FilesControl/fileOld.txt ]]; then
            cd ~/FilesControl
            echo "You are in “FilesControl”."
            touch fileOld.txt
        else
            cd ~/FilesControl
            echo "You are in “FilesControl”."
        fi

        sudo apt install tree
        tree -a > fileLast.txt
        echo Files results were written to “fileLast.txt”.
        [ -s fileOld.txt ]

        echo $? > temp
        if [[ $(< temp) == "1" ]]; then
            printf "There is no old record for files comparation.\n"
```

```
        cat fileLast.txt > fileOld.txt
        echo "Files results were copied to “fileOld.text” for future comparisons."
    elif cmp -s fileOld.txt fileLast.txt; then
        printf "There is no change on files.\n"
    else
        printf 'There are changes on files!\n\n'
        diff fileOld.txt fileLast.txt > results
        echo "Differences between old and last results are written to “results”."
        diff fileOld.txt fileLast.txt
        cat fileLast.tx > t fileOld.txt
        echo "Files results were copied to “fileOld.txt” from 'fileLast.txt'".
    fi
    rm temp
}

function controlTasks () {
    cd ~
    if [[ ! -d ~/TasksControl ]]; then
        echo creating a directory, “TasksControl”, to the home…
        mkdir ~/TasksControl
        cd ~/TasksControl
        echo "You are in “TasksControl”."
        touch taskOld.txt
    elif [[ ! -f ~/TasksControl/taskOld.txt ]]; then
        cd ~/TasksControl
        echo "You are in “TasksControl”."
        touch taskOld.txt
    else
        cd ~/TasksControl
        echo "You are in “TasksControl”."
    fi

    crontab -l > taskLast.txt
    echo Tasks results were written to “taskLast.txt”.
    [ -s taskOld.txt ]
    echo $? > temp

    if [[ $(< temp) == "1" ]]; then
        printf "There is no old record for Tasks comparation.\n"
        cat taskLast.txt > taskOld.txt
        echo "Tasks results were copied to “taskOld.text” for future comparisons."
    elif cmp -s taskOld.txt taskLast.txt; then
        printf "There is no change on Tasks.\n"
    else
```

```
            printf 'There are changes on Tasks!\n\n'
            diff taskOld.txt taskLast.txt > results
            echo "Differences between old and last results are written to “results”."
            diff taskOld.txt taskLast.txt
            cat taskLast.txt > taskOld.txt
            echo "Tasks results were copied to “taskOld.txt” from 'taskLast.txt'".
        fi
        rm temp
}

controlAccounts
CPUBenchmark
memoryBenchmark
controlDiskSpace
controlProcess
controlServices
controlFiles
controlTasks
```

3 Results

- Modifying critical files in the system can create problems in the operation of the system.
- Being a superuser is as dangerous as it is comfortable.
- Using Apache, OpenSSH and Vsftpd services allows attacks that can be performed remotely. It is possible to make the services more secure by changing the configuration files.
- System security can be improved with simple changes.
- Thanks to the log files, we can see any changes made in the system.
- Keeping our operating system and the packages we use increases security.

4 Conclusion

I learned a lot about user types, how to install and test Apache, OpenSSH and Vsftpd services, how to make them safer, where system and application logs are kept, which system files are critical, and system security. I gained experience on these issues. During my research, I also learned about topics that do not have answers to assignments directly.

5 Grade

I could not perform the "unusual log entries" step in assignment 9 because I could not decide which log file to import. I did not do the "unusual network usage" step of the same assignment because I could not find how to do it on the terminal.

I tried to fulfill all other tasks completely. I don't think I am very successful in the 9th and 10th assignments, but I tried to do my best within the circumstances. I took care to indicate the sources I used in my answers. There were places where I made copy / paste for definitions. But I did not include any information I do not understand in my report. I believe I will get the score I deserve.

6 References

[1] https://mediatemple.net/community/products/dv/204643890/an-introduction-to-the-root-user

[2] https://www.omnisecu.com/gnu-linux/redhat-certified-engineer-rhce/introduction-to-linux-user-administration.php

[3] https://phoenixnap.com/kb/how-to-install-apache-web-server-on-ubuntu-18-04

[4] https://tipsonubuntu.com/2019/10/17/enable-secure-shell-ssh-service-ubuntu-19-10/

[5] https://www.howtoforge.com/tutorial/ubuntu-vsftpd/

[6] https://www.tecmint.com/apache-security-tips/

[7] https://help.ubuntu.com/community/SSH/OpenSSH/Configuring

[8] https://www.vultr.com/docs/how-to-secure-vsftpd-with-ssl-tls

[9] https://ubuntu.com/tutorials/viewing-and-monitoring-log-files#2-log-files-locations

[10] https://www.tecmint.com/linux-directory-structure-and-important-files-paths-explained/

[11] http://manpages.ubuntu.com/manpages/bionic/man7/inotify.7.html#colophon

[12] https://www.linuxjournal.com/content/linux-filesystem-events-inotify

[13] https://www.digitalocean.com/community/tutorials/how-to-monitor-system-authentication-logs-on-ubuntu

[14] https://www.cyberciti.biz/faq/linux-unix-openssh-block-root-user/

[15] https://linuxconfig.org/how-to-benchmark-your-linux-system