Secure Systems Administration Assignment

Daniel Gallagher

L00158616

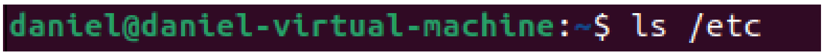
L00158616@atu.ie

11th December 2022

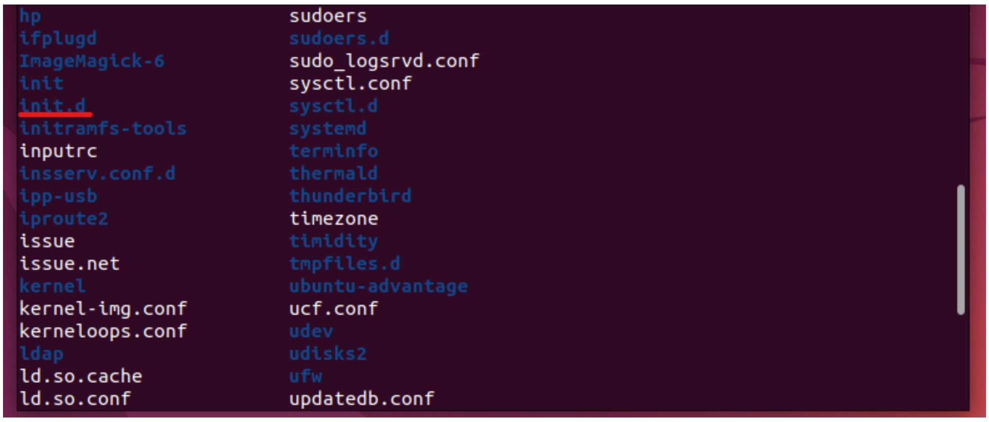
I am using ubuntu version 22.04.1 LTS for the Assignment.

Question 1

A.

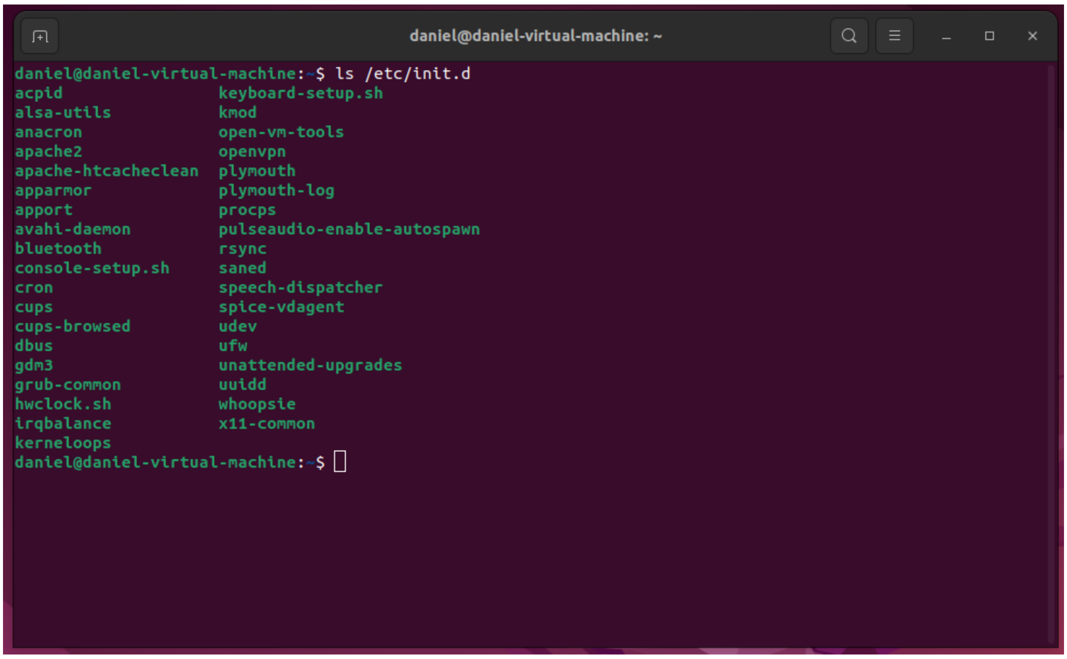


```
daniel@daniel-virtual-machine:~$ ls /etc
```



```
hp                      sudoers
ifplugd                 sudoers.d
ImageMagick-6           sudo_logsrvd.conf
init                    sysctl.conf
init.d                  sysctl.d
initramfs-tools         systemd
inputrc                 terminfo
insserv.conf.d          thermald
ipp-usb                 thunderbird
iproute2                timezone
issue                   timidity
issue.net               tmpfiles.d
kernel                  ubuntu-advantage
kernel-img.conf         ucf.conf
kerneloops.conf         udev
ldap                    udisks2
ld.so.cache             ufw
ld.so.conf              updatedb.conf
```

- The start-up scripts are in the /etc/init.d directory.



```
daniel@daniel-virtual-machine:~$ ls /etc/init.d
acpid                  keyboard-setup.sh
alsa-utils             kmod
anacron                open-vm-tools
apache2                openvpn
apache-htcacheclean    plymouth
apparmor               plymouth-log
apport                 procps
avahi-daemon           pulseaudio-enable-autospawn
bluetooth              rsync
console-setup.sh       saned
cron                   speech-dispatcher
cups                   spice-vdagent
cups-browsed           udev
dbus                   ufw
gdm3                   unattended-upgrades
grub-common            uuidd
hwclock.sh             whoopsie
irqbalance             x11-common
kerneloops
daniel@daniel-virtual-machine:~$ 
```

/etc/init.d/apache2 – It can be used to start, stop, and restart the Apache server, as well as to configure various aspects of the server's behaviour.

/etc/init.d/cron – It can be used to start, stop, and restart the cron daemon, as well as to configure how the daemon behaves.

/etc/init.d/rsync - It can be used to start, stop, and restart the rsync service, as well as to configure how the service behaves.

/etc/init.d/ufw - It can be used to start, stop, and restart the ufw service, as well as to enable or disable specific firewall rules.

/etc/init.d/acpid - It can be used to start, stop, and restart the acpid daemon, as well as to configure how the daemon behaves.

B.

Describing the Samba Server

• Samba is an open-source implementation of the SMB/CIFS network protocol, which allows Linux and other Unix-like operating systems to share files, printers, and other resources with Windows-based systems. It provides access to shared resources over the network using the SMB/CIFS protocol.

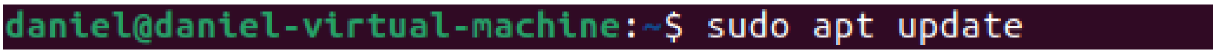
Its Uses

• Samba can be used in a variety of ways, including to provide file and printer sharing services on a network, to allow Linux systems to access resources on a Windows-based network, or to enable Linux systems to participate in a Windows domain.

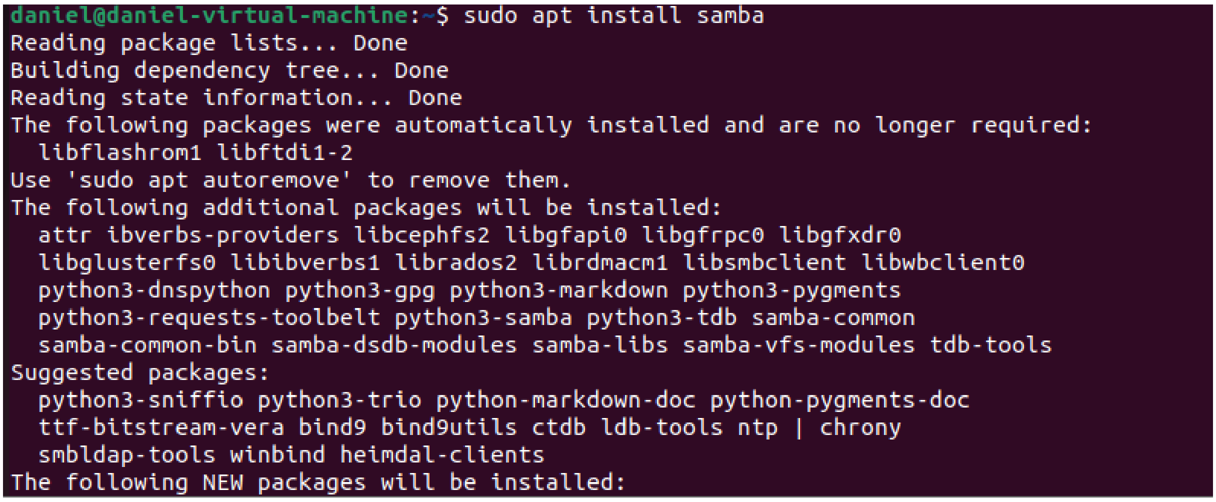
• The Samba server can also be configured to act as a domain controller for a Windows domain, allowing Linux systems to be managed and authenticated in the same way as Windows systems.

C.

Installing Samba

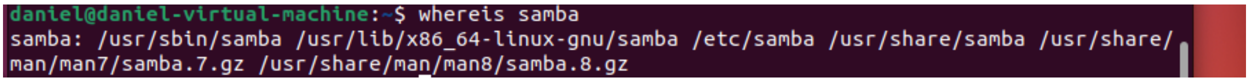


```
daniel@daniel-virtual-machine:~$ sudo apt update
```



```
daniel@daniel-virtual-machine:~$ sudo apt install samba
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  attr ibverbs-providers libcephfs2 libgfapi0 libgfrpc0 libgfxdr0
  libglusterfs0 libibverbs1 librados2 librdmacm1 libsmbclient libwbclient0
  python3-dnspython python3-gpg python3-markdown python3-pygments
  python3-requests-toolbelt python3-samba python3-tdb samba-common
  samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
Suggested packages:
  python3-sniffio python3-trio python-markdown-doc python-pygments-doc
  ttf-bitstream-vera bind9 bind9utils ctdb ldb-tools ntp | chrony
  smbldap-tools winbind heimdal-clients
The following NEW packages will be installed:
```

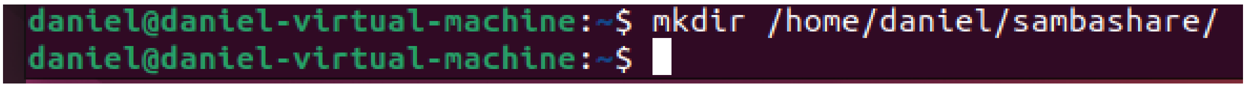
• To install Samba, I ran the sudo apt update command and the sudo apt install samba command.



```
daniel@daniel-virtual-machine:~$ whereis samba
samba: /usr/sbin/samba /usr/lib/x86_64-linux-gnu/samba /etc/samba /usr/share/samba /usr/share/
man/man7/samba.7.gz /usr/share/man/man8/samba.8.gz
```

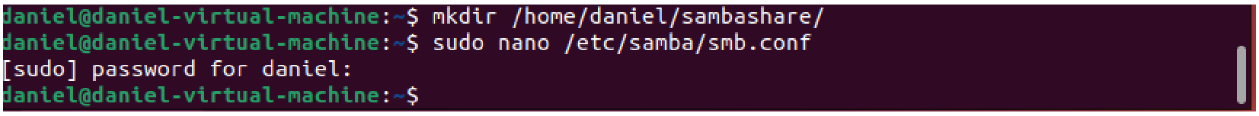
- I checked if the installation was successful by running the command whereis samba.

Setting up Samba



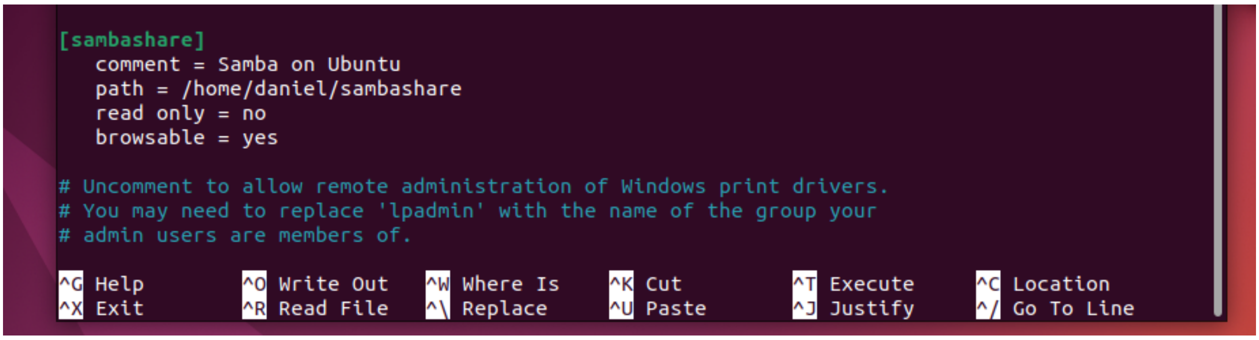
```
daniel@daniel-virtual-machine:~$ mkdir /home/daniel/sambashare/
daniel@daniel-virtual-machine:~$ 
```

- I created a new folder called sambashare in my home directory.



```
daniel@daniel-virtual-machine:~$ mkdir /home/daniel/sambashare/
daniel@daniel-virtual-machine:~$ sudo nano /etc/samba/smb.conf
[sudo] password for daniel:
daniel@daniel-virtual-machine:~$
```

- to add the new directory as a share, I edit the configuration file for samba /etc/samba/smb.conf

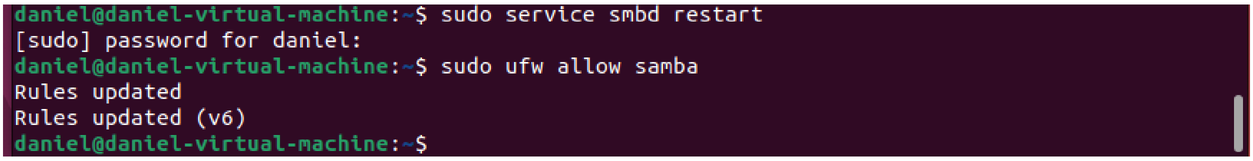


```
[sambashare]
   comment = Samba on Ubuntu
   path = /home/daniel/sambashare
   read only = no
   browsable = yes

# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

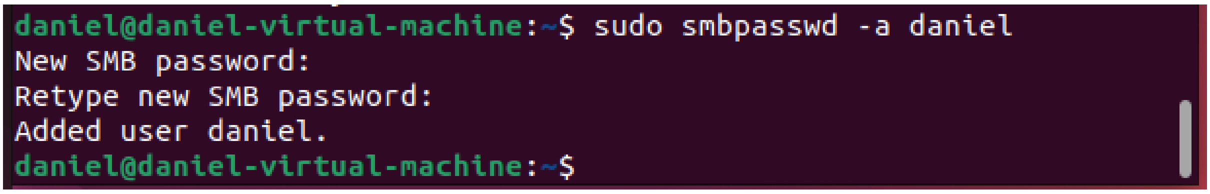
- the image above shows configuration for the new share which has a path with the directory of my share, a read only to modify the contents of the share folder and a browsable which will allow file managers to list the share under Network.



```
daniel@daniel-virtual-machine:~$ sudo service smbd restart
[sudo] password for daniel:
daniel@daniel-virtual-machine:~$ sudo ufw allow samba
Rules updated
Rules updated (v6)
daniel@daniel-virtual-machine:~$
```

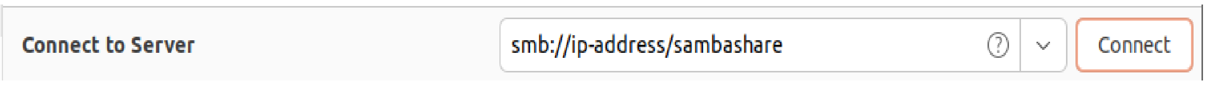
- Saving and restarting Samba for the new share to take effect and updating the firewall rules to allow samba traffic.

Setting up a User Account and Connecting to Share

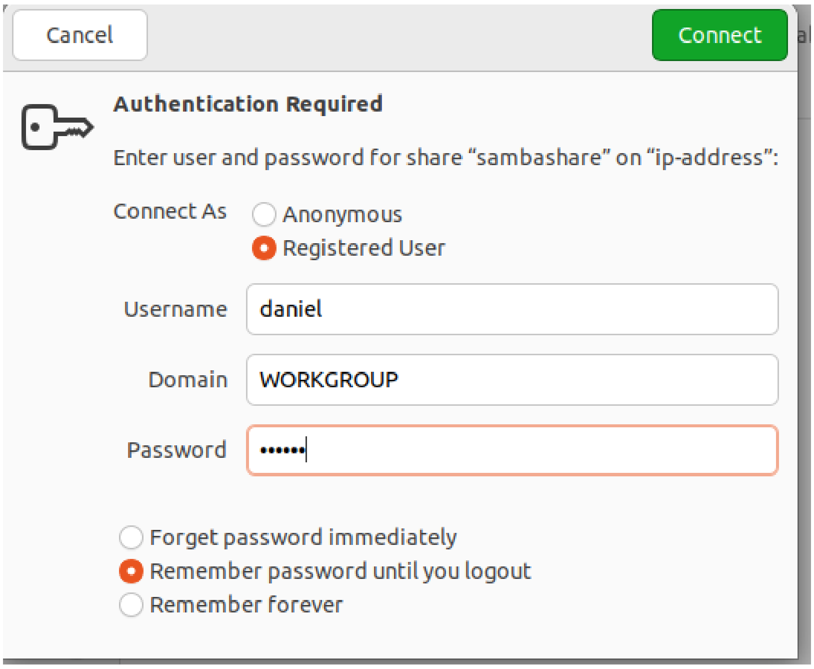


```
daniel@daniel-virtual-machine:~$ sudo smbpasswd -a daniel
New SMB password:
Retype new SMB password:
Added user daniel.
daniel@daniel-virtual-machine:~$
```

- Setting up a samba password for my user account



Connect to Server smb://ip-address/sambashare



Authentication Required

Enter user and password for share “sambashare” on “ip-address”:

Connect As ○ Anonymous ● Registered User

Username: daniel

Domain: WORKGROUP

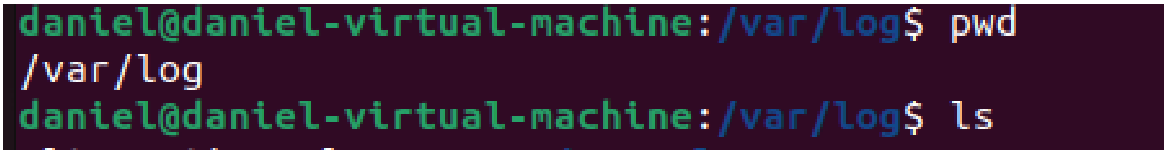
Password: ••••••

○ Forget password immediately
● Remember password until you logout
○ Remember forever

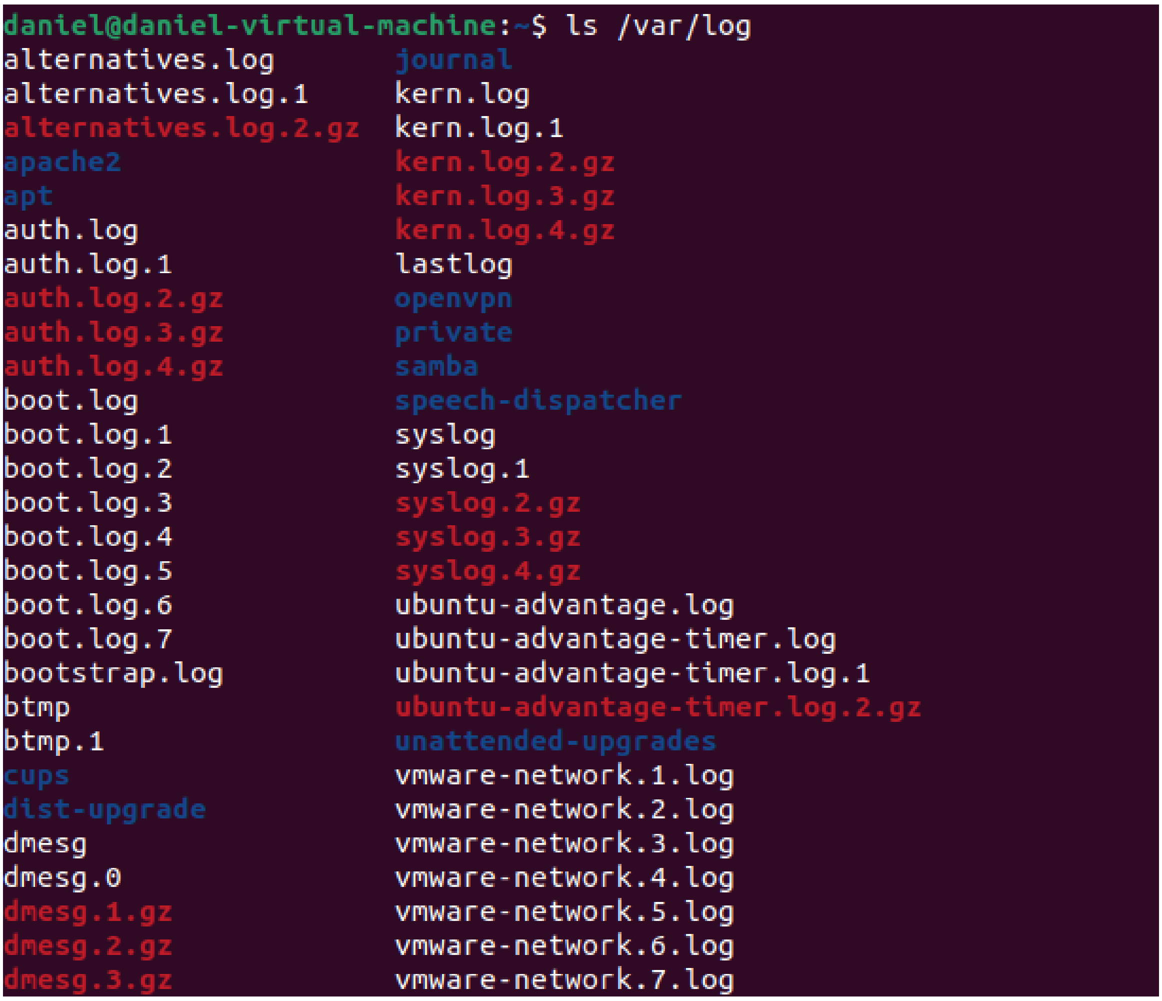
- Connecting to Share.

Question 2

A.



```
daniel@daniel-virtual-machine:/var/log$ pwd
/var/log
daniel@daniel-virtual-machine:/var/log$ ls
```



```
daniel@daniel-virtual-machine:~$ ls /var/log
alternatives.log       journal
alternatives.log.1     kern.log
alternatives.log.2.gz  kern.log.1
apache2                kern.log.2.gz
apt                    kern.log.3.gz
auth.log               kern.log.4.gz
auth.log.1             lastlog
auth.log.2.gz          openvpn
auth.log.3.gz          private
auth.log.4.gz          samba
boot.log               speech-dispatcher
boot.log.1             syslog
boot.log.2             syslog.1
boot.log.3             syslog.2.gz
boot.log.4             syslog.3.gz
boot.log.5             syslog.4.gz
boot.log.6             ubuntu-advantage.log
boot.log.7             ubuntu-advantage-timer.log
bootstrap.log          ubuntu-advantage-timer.log.1
btmp                   ubuntu-advantage-timer.log.2.gz
btmp.1                 unattended-upgrades
cups                   vmware-network.1.log
dist-upgrade           vmware-network.2.log
dmesg                  vmware-network.3.log
dmesg.0                vmware-network.4.log
dmesg.1.gz             vmware-network.5.log
dmesg.2.gz             vmware-network.6.log
dmesg.3.gz             vmware-network.7.log
```

B.

- The logs folder contains log files that record important events and messages from the operating system and various applications.

Some examples of log files you will find in the log's directory include:

auth.log - This file contains log messages related to authentication, such as when users log in and out of the system, and when authentication failures occur.

syslog - This file contains system-wide log messages, including information about the kernel, system services, and other system-level components.

mail.log - This file contains log messages related to the mail system, including information about incoming and outgoing messages, as well as any errors or warnings that occur during the mail process.

dmesg - This file contains log messages generated by the kernel, including information about hardware devices and drivers that are loaded during the boot process.

C.

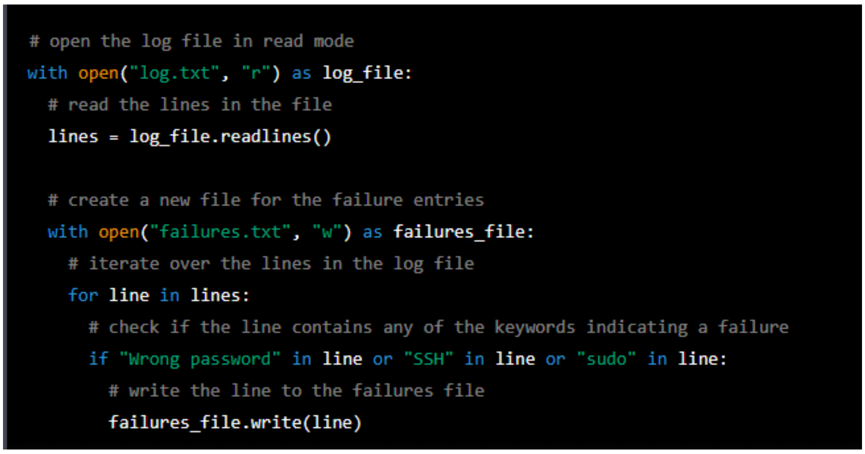
grep - you could use the grep command to search for log entries that contain a specific keyword, or that were generated by a specific application.

awk - you could use the awk command to extract the timestamp, severity level, and message text from each log entry, and then to sort or filter the entries based on those values.

tac – you could use the tac command if you want to view the most recent log entries without having to scroll to the end of the file.

sed – you could use sed to allow you to manipulate log entries in various ways, such as by replacing specific words or phrases, or by deleting certain lines or entries.

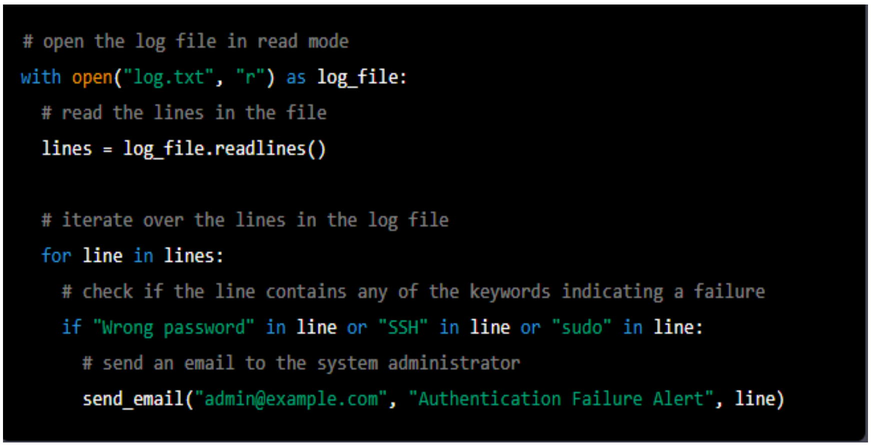
D.



```python
# open the log file in read mode
with open("log.txt", "r") as log_file:
  # read the lines in the file
  lines = log_file.readlines()

  # create a new file for the failure entries
  with open("failures.txt", "w") as failures_file:
    # iterate over the lines in the log file
    for line in lines:
      # check if the line contains any of the keywords indicating a failure
      if "Wrong password" in line or "SSH" in line or "sudo" in line:
        # write the line to the failures file
        failures_file.write(line)
```

E.

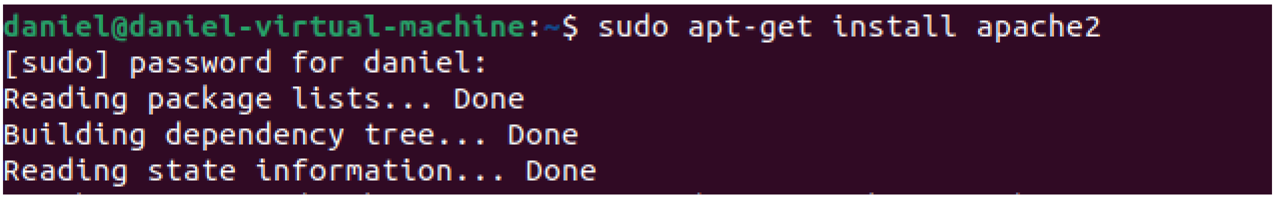


```python
# open the log file in read mode
with open("log.txt", "r") as log_file:
  # read the lines in the file
  lines = log_file.readlines()

  # iterate over the lines in the log file
  for line in lines:
    # check if the line contains any of the keywords indicating a failure
    if "Wrong password" in line or "SSH" in line or "sudo" in line:
      # send an email to the system administrator
      send_email("admin@example.com", "Authentication Failure Alert", line)
```

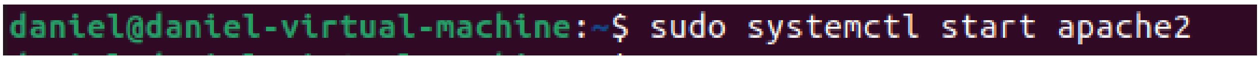
Question 3

A.



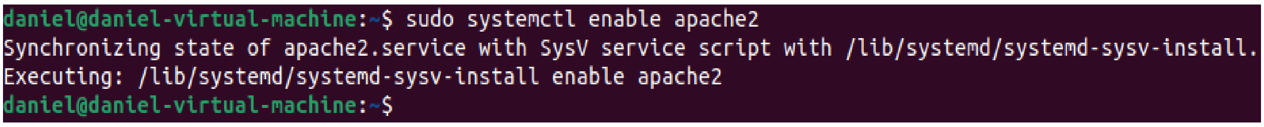
```
daniel@daniel-virtual-machine:~$ sudo apt-get install apache2
[sudo] password for daniel:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

- I used the sudo apt-get install apache2 command to install Apache.



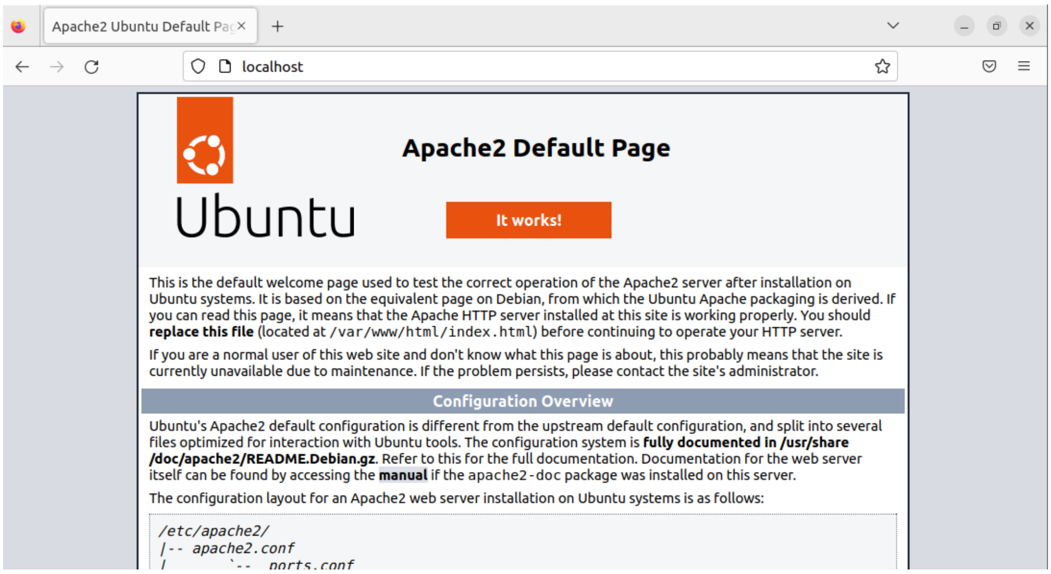
```
daniel@daniel-virtual-machine:~$ sudo systemctl start apache2
```

- I used the sudo systemctl start apche2 command to start the Apache service.



```
daniel@daniel-virtual-machine:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
daniel@daniel-virtual-machine:~$
```

- I used the sudo systemctl enable apache2 command so that Apache automatically starts when the system boots up.



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

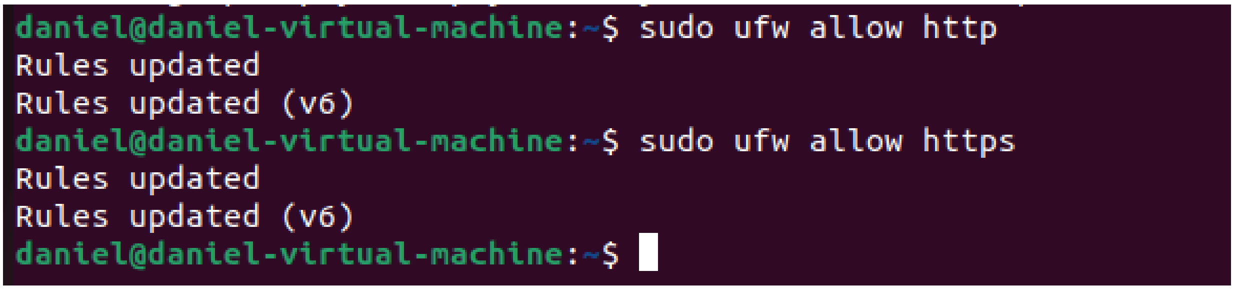
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
```

- Verifying that Apache is running by visiting http://localhost in Firefox and it shows the Apache default page.



```
daniel@daniel-virtual-machine:~$ sudo ufw allow http
Rules updated
Rules updated (v6)
daniel@daniel-virtual-machine:~$ sudo ufw allow https
Rules updated
Rules updated (v6)
daniel@daniel-virtual-machine:~$ 
```

- To allow incoming traffic to my Apache server, I adjusted my firewall settings and used sudo ufw allow http command and the sudo ufw allow https command to allow HTTP and HTTPS traffic.

B.

The different ways that can be used to secure an Apache webserver are:

1. Is to configure the Apache webserver to use SSL/TLS encryption to secure communication between the server and clients.

2. Is to restrict access to the server by blocking unnecessary ports and using a firewall.

3. Is to enable the Apache webserver's built-in security features, such as mod security and mod evasive.

4. Is to keep the Apache webserver and the operating system up to date with the latest security patches.

5. Is to Implement a backup and a disaster recovery plan to ensure that you can recover from any security breaches or other disasters

6. Is to use intrusion detection and prevention systems to monitor for and protect against malicious activity.