Phase 1: System Planning and Distribution Selection (Week 1)

1. Create a System Architecture Diagram showing both systems and network connections?

Ans: An administrator workstation and a headless Ubuntu Server are each of the separate systems that make up the client-server architecture used for operating system deployment. An Ubuntu Server hosts platform utilities, or an admin server utilises it in remote support. A Virtual Box Host Only personal network allows the systems to link using one another while still free of other networks. The current Secure Shell (SSH), which protects the entire process, is used for running the systems online. The computer running Ubuntu has been given the private IP address 192.168.56.101 for the connection between users and the server.

2. Distribution Selection Justification comparing your chosen server distribution with alternatives?

Ans: Ubuntu Server 24.04 Long- Term Support (LTS) is an OS the fact was chosen for usage on the project because its support, capacity, and control were taken into factors. The Ubuntu Long Term Support (LTS) operating system has a long support term. On stability, one of the strengths of Ubuntu Server is that its foundation is on the Debian platform, which is dependent on newer and proven software. This is achieved through a controlled release by Canonical. This is remarkably beneficial to admins since they will be able to reap the benefits of newer technological developments without having an adverse effect on the stability of the actual server.

At the forefront of Debian is ensuring that its server is extremely stable. This is attained by ensuring that software is frozen. This has, at times, caused users to possess outdated software. Another option was also CentOS, which was also taken as an alternative. CentOS had generally always been widely used because of its ability to provide compatibility for Red Hat Enterprise Linux, commonly known as RHEL. However, due to the emerging changes in the release of CentOS, known as CentOS Stream, it has not yet reached its peak in terms of predictability.

There are also emerging alternatives that could potentially replace CentOS, which include Alma Linux and Rocky Linux, but compared to Ubuntu Server, community resources are not elaborate. Ubuntu Server is well-documented, and its community is loyal and very popular within the industry. Ubuntu Server is also well-supported by modern security software and very popular within academic and cloud computing settings because of its high compatibility with security software and virtualization software. As far as operating systems are concerned, the most appropriate one that can serve as an example in the critical areas involved in the workings of an OS, which are process scheduling, memory management, file system structure, networking, and access control mechanism, is Ubuntu Server OS. Ubuntu Server OS is the most appropriate and genuine one among all other server versions available for selection in terms of functionality and compatibility with stability and the latest version.

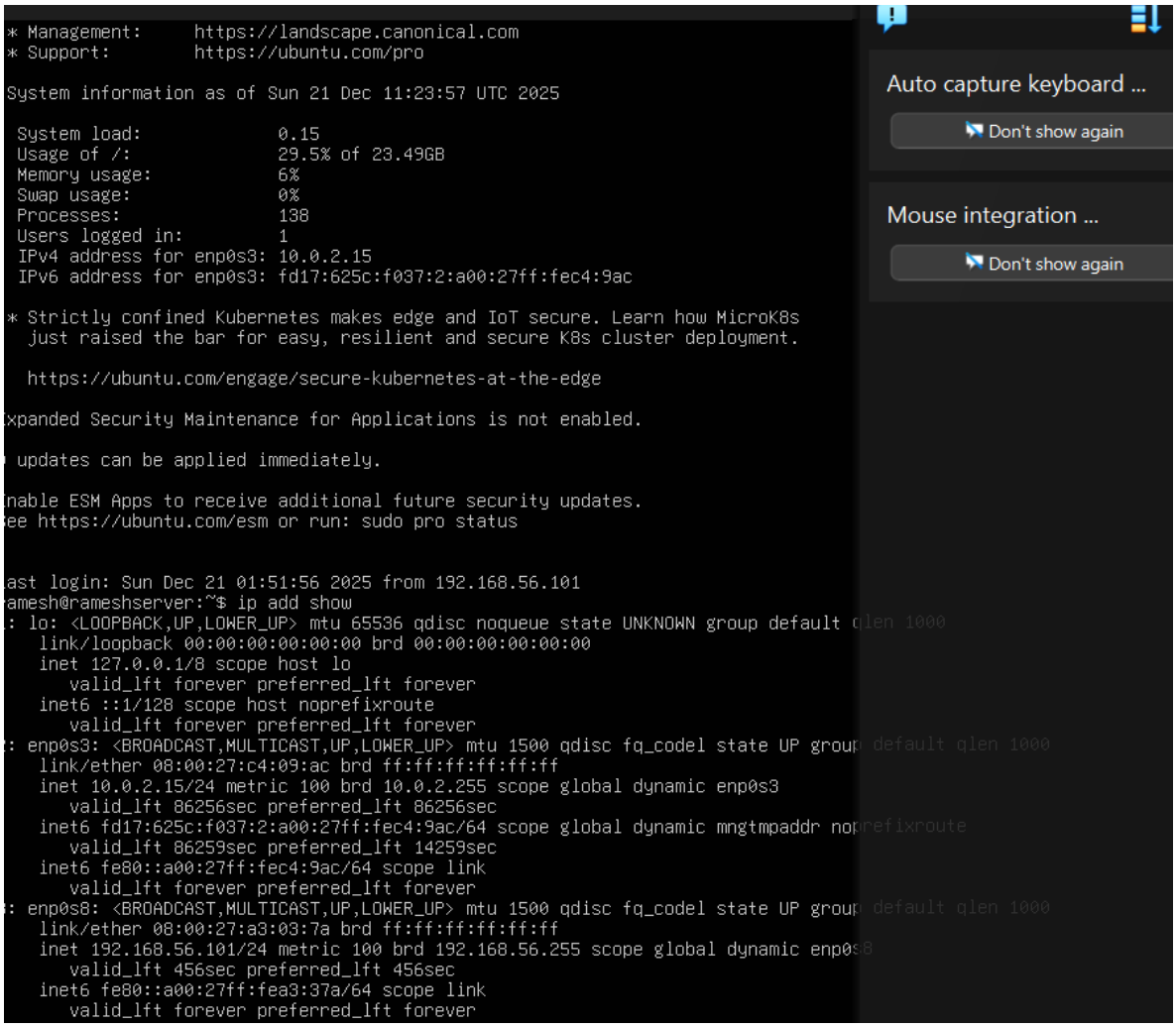
3.Workstation configuration decision justifying your choice of workstation option?

Ans: A different workstation was selected for computer management tasks to implement guidelines with control of the system. By removing the need to run a GUI from the server, controlling it via SSH decreases the server's power needs while leaving it less open to attacks.

The workstation provides a setting for safely entering commands, checking performance, and managing file systems. Each of those acts complies with the least rights ideal. Working on a workstation highlights commands ability and remote workstation operation, both of which are key capacities as will be learnt during this module.

4. Network configuration documentation covering VirtualBox settings and IP addressing?

Ans:



The network setup uses VirtualBox Host-Only Networking to make a private network between the workstation and the Ubuntu Server. This keeps the server off the open internet. 192.168.56.101 is the private IP address of the Ubuntu server on the private network. This device connects to the VirtualBox host-only

network, creating safe interactions between the server while the owner's machine. This setup shows how the operating system handles network connections, IP addresses, and getting data where it needs to go in a virtual environment, all while staying secure by keeping things separate

5. Using a CLI, document system specifications using `uname`, `free`, `df -h`, `ip addr`, and `lsb_release`?

Uname -a:

The uname -a search term will be used to obtain more details about the operating system of the computer. The output shows the system is using a 64-bit CPU and most fresh operating system build used by Ubuntu. Also, it shows that the kernel supports changing advance and parallel processing, both of which improve system speed and performance. The IP address that shows confirms that the commands are running on the right server.

Free -h:

The system's memory usage is shown in a readable format by the free -h command. just some the server's 3.8 GB of virtual memory is currently in use, relating to the output. The system is fully loaded and functioning effectively because most memory is still available. with the low demand on memory, it may be correct that no spare room is set up.

df -h:

```
ramesh@rameshserver:~$ df -h
Filesystem                         Size  Used Avail Use% Mounted on
tmpfs                              392M  1.1M  391M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv   24G  7.1G   16G  32% /
tmpfs                              2.0G     0  2.0G   0% /dev/shm
tmpfs                              5.0M     0  5.0M   0% /run/lock
/dev/sda2                          2.0G  197M  1.6G  11% /boot
tmpfs                              392M   12K  392M   1% /run/user/1000
```

Ip addr:

```
amesh@rameshserver:~$ ip add show
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
```

The server is set to utilise multiple network interfaces, as confirmed by the ip address display command. Internal system communication uses the return interface, and enp0s3 uses a firewall to offer internet access. An IP address that is secret is provided the enp0s8 network interface, who is utilised for safe SSH communication with the host computer. Performance and danger are made better by this online division.

lsb_release -a:

```
ramesh@rameshserver:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.3 LTS
Release:        24.04
Codename:       noble
```

Linux's release specifics may be seen using a the lsb_release command. The Linux Standards Basis, or the LSB over shorter, sets uniform norms across all Linux distributions. The operating system version, release number, and password that are currently operating on the server is confirmed by this command.

Phase 2: Security Planning and Testing Methodology (Week 2):

1. Create a performance testing plan describing your remote monitoring methodology and testing approach?

Ans: for keeping track of the Ubuntu Operating System servers' actions, you setup that to a practice plan. We can use a safe shell to log to via the internet. With SSH, servers connect to a secure network (enp0s8, IP address 192.168.56.101). So, no need to watch the screen. When the computer is operating smoothly, we monitor its CPU, memory, disc space, or networks. when our team apply safety steps, we'll observe how these values shift. This remote viewing method is like how servers are maintained. By doing this, we guarantee that any system hits via the privacy settings are detected and measured.

The testing approach involves:

. Watching system resource use when the system is running normally.

. Finding possible drops in performance caused by security measures.

. Making sure the system remains stable when managed remotely.

. This method helps with numerical assessment and gives proof for later choices about how to improve the system.

2. Security Configuration Checklist covering SSH hardening, firewall configuration, mandatory access control, automatic updates, user privilege management, and network security?

Ans: Security Configuration Checklist are:

1 .SSH hardening:

. Turn off root login for SSH.

. Use SSH keys instead of passwords.

. Only let SSH work on your private network (enp0s8).

2.firewall configuration:

. Turn on a firewall on your machine.

. Only allow needed services like SSH.

. Block everything else coming in.

3. mandatory access control:

. Turn on Appar Mòr to control what apps can do.

. Just stick with the default Ubuntu security settings to keep things safe.

4. automatic updates:

. Turn on automatic security updates.

. This helps patch security holes fast.

5. user privilege management:

. Don’t use the root account for everyday stuff.

. Use Sudo carefully to grant admin rights when required

6. network security:

. use private IPs for your admin stuff.

. Keep things separate using VirtualBox host-only networking.

. Don't directly expose things to the outside internet.

3. Threat Model identifying at least 3 specific security threats with mitigation strategies?

Ans: The threat identifying at least 3 specific security threat with mitigation strategies are as follows:

Attacks that try to guess SSH login info are a serious security risk. Possible fixes are using SSH keys and disabling password logins.

Unauthorized network access, or external access to services, is a medium-level threat that can be addressed through firewall rules and network isolation.

Privilege escalation, which occurs when elevated permissions are misused, is another medium-level threat best handled by using non-root users and implementing Sudo policies.

This security model makes security measures directly related to the system's main risks.

Phase 3: Application Selection for Performance Testing (Week 3):

1. Select applications representing different workload types (e.g. CPU-intensive, RAM intensive, I/O-intensive, Network-intensive, and Server applications such as game servers) for performance evaluation and create an Application Selection Matrix listing applications with justifications for choosing them.?

Ans: The select application representing different workload types are follows:

Workload type	Selected Application	Justification
CPU-intensive	Stress-ng	used to assess CPU capacity and for plans as

		maintaining a heavy CPU use.
RAM intensive	Memory test	offers controlled memory allocation for testing memory availability and management.
i/O-intensive	fio	I create disc read/write operations to examine disc I/O and archive memory speed.
Network-intensive	Iperf3	evaluates the timing and speeds of the network.
Sever application	nginx	Correct server workloads are simulated by a lightweight web server.

These applications were picked because they are common, simple, and work well for testing in a virtual Ubuntu Server setup.

2. Installation Documentation with exact commands for SSH-based installation?

Ans:

1.Sudo apt update:

```
ramesh@rameshserver:~$ sudo apt update
[sudo] password for ramesh:
Hit:1 http://gb.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://gb.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://gb.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

The computer's software listing was revised using a command like Sudo apt update. The output shows that the server's packages are current, showing that it is ready for app installs, and that the Linux sources was visited.

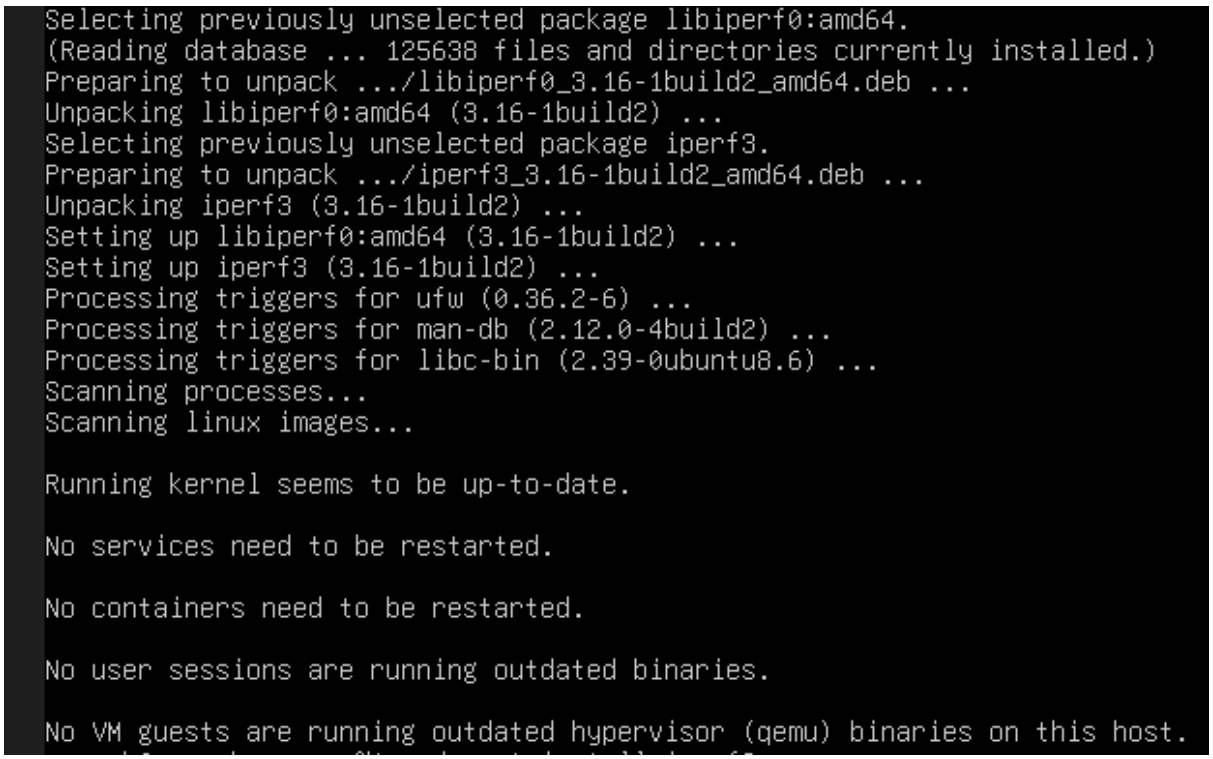
Sudo apt install stress-ng -y:

Installing the stress-ng package took place via the Ubuntu package manager. The output confirms which the present version is the most modern one and it says the archive arrives and working.

Sudo apt install fio -y:

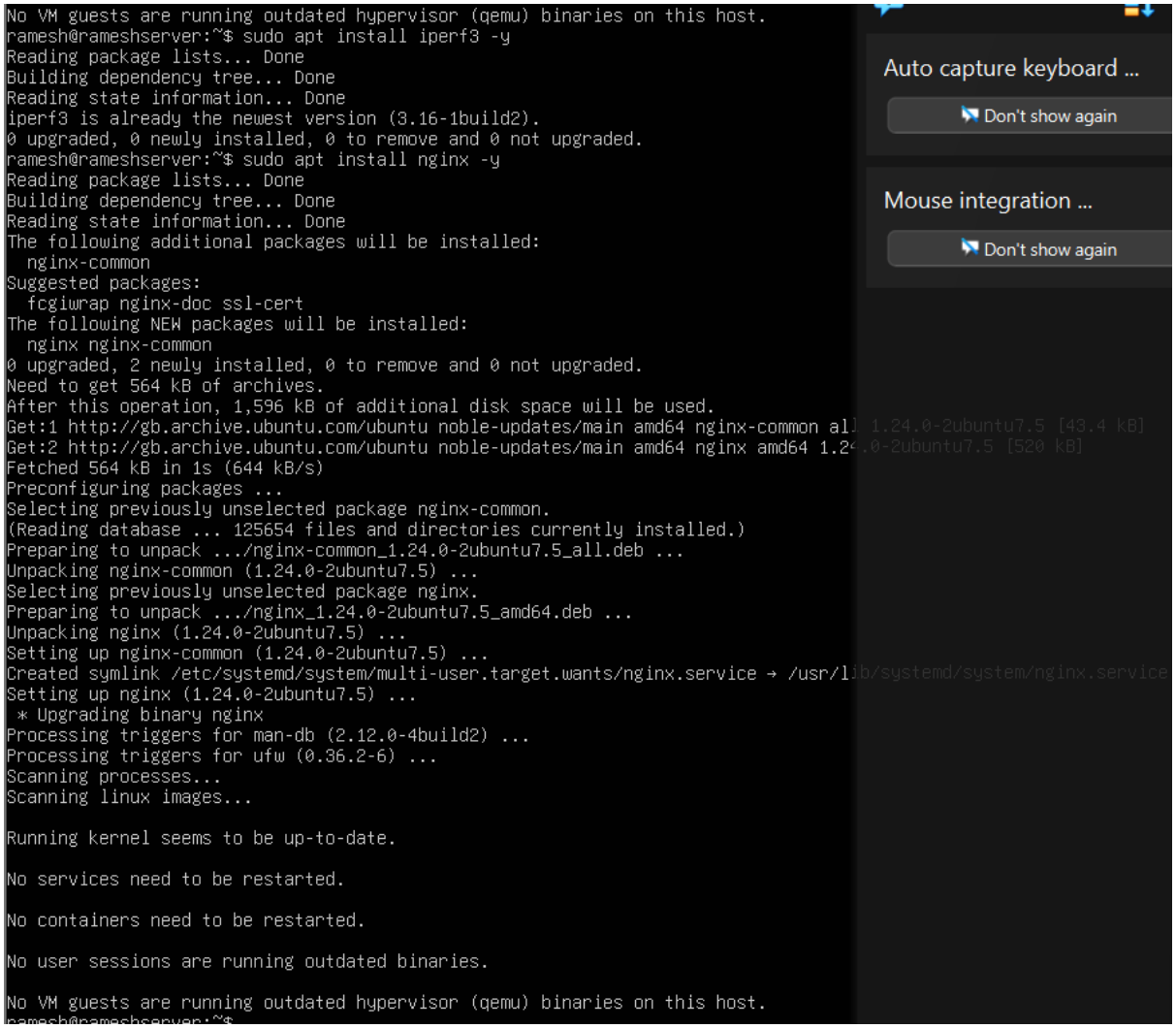
```
Preparing to unpack .../10-librdmacm1t64_50.0-2ubuntu0.2_amd64.deb ...
Unpacking librdmacm1t64:amd64 (50.0-2ubuntu0.2) ...
Selecting previously unselected package librados2.
Preparing to unpack .../11-librados2_19.2.3-0ubuntu0.24.04.1_amd64.deb ...
Unpacking librados2 (19.2.3-0ubuntu0.24.04.1) ...
Selecting previously unselected package libpmemobj1:amd64.
Preparing to unpack .../12-libpmemobj1_1.13.1-1.1ubuntu2_amd64.deb ...
Unpacking libpmemobj1:amd64 (1.13.1-1.1ubuntu2) ...
Selecting previously unselected package librbd1.
Preparing to unpack .../13-librbd1_19.2.3-0ubuntu0.24.04.1_amd64.deb ...
Unpacking librbd1 (19.2.3-0ubuntu0.24.04.1) ...
Selecting previously unselected package fio.
Preparing to unpack .../14-fio_3.36-1ubuntu0.1_amd64.deb ...
Unpacking fio (3.36-1ubuntu0.1) ...
Setting up libboost-thread1.83.0:amd64 (1.83.0-2.1ubuntu3.1) ...
Setting up libnbd0 (1.20.0-1) ...
Setting up libglusterfs0:amd64 (11.1-4ubuntu0.1) ...
Setting up libboost-iostreams1.83.0:amd64 (1.83.0-2.1ubuntu3.1) ...
Setting up libdaxctl1:amd64 (77-2ubuntu2) ...
Setting up libndctl6:amd64 (77-2ubuntu2) ...
Setting up librdmacm1t64:amd64 (50.0-2ubuntu0.2) ...
Setting up libpmem1:amd64 (1.13.1-1.1ubuntu2) ...
Setting up libgfxdr0:amd64 (11.1-4ubuntu0.1) ...
Setting up librados2 (19.2.3-0ubuntu0.24.04.1) ...
Setting up libpmemobj1:amd64 (1.13.1-1.1ubuntu2) ...
Setting up librbd1 (19.2.3-0ubuntu0.24.04.1) ...
Setting up libgfrpc0:amd64 (11.1-4ubuntu0.1) ...
Setting up libgfapi0:amd64 (11.1-4ubuntu0.1) ...
Setting up fio (3.36-1ubuntu0.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
```

Sudo apt install iperf3 -y:



Using the Ubuntu package managers, Iperf3 was installed. The output shows that all required interactions were properly opened and setup, and no computer services needed to be resumed after the install.

Sudo apt install nginx -y:



3. Expected Resource Profiles documenting anticipated resource usage?

Ans: Estimated Use of Resources

stressing (CPU tests):

This is meant to improve CPU usage on one or more cores. Testing planning and the system's ability to manage high loads is the aim.

stressful (tests of memory):

To test memory planning, caching, and if memory is available since needed, this requires significant amounts of RAM.

Fio:

This is meant to put a lot of reads and writes on the disc. It will check the file system's operation.

Iperf3:

This should boost the volume of data passing in the network, enabling tests of delay, speed, and data packet handling.

Expected to consume moderate CPU and memory resources while processing the requests for simulating actual server scenarios.

These profiles create a minimum standard for future comparison.

4.Monitoring Strategy explaining measurement approach for each application?

Ans: System performance can be remotely viewed using SSH command-line tools. The monitoring approach that will be followed is:

CPU Usage: Used to analyse the system's scheduling activities and processing workload during CPU-intensive tasks.

Memory Usage: Monitored to gauge memory pressure and availability under RAM-intensive workloads.

Disk Activity: Analysed during disk I/O benchmarking for reading or writing performance.

Network Performance: Tested on the network to measure its bandwidth and stability.

Overall System Load: Tested in the server application environment for performance measurement.

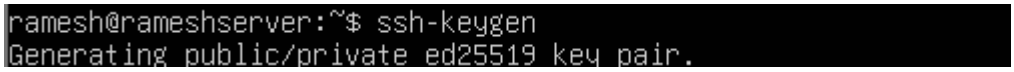
Measurements will be taken before, during, and after the execution of the workload to enable comparison and analysis.

Phase 4: Initial System Configuration & Security Implementation (Week 4):

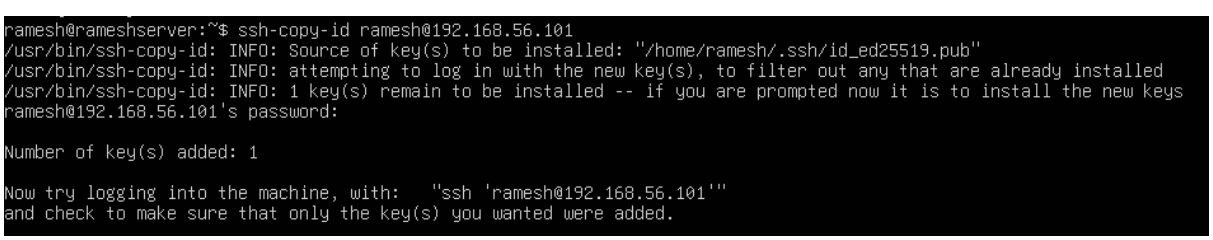
1.Configure SSH with key-based authentication

Step performed (via SSH)

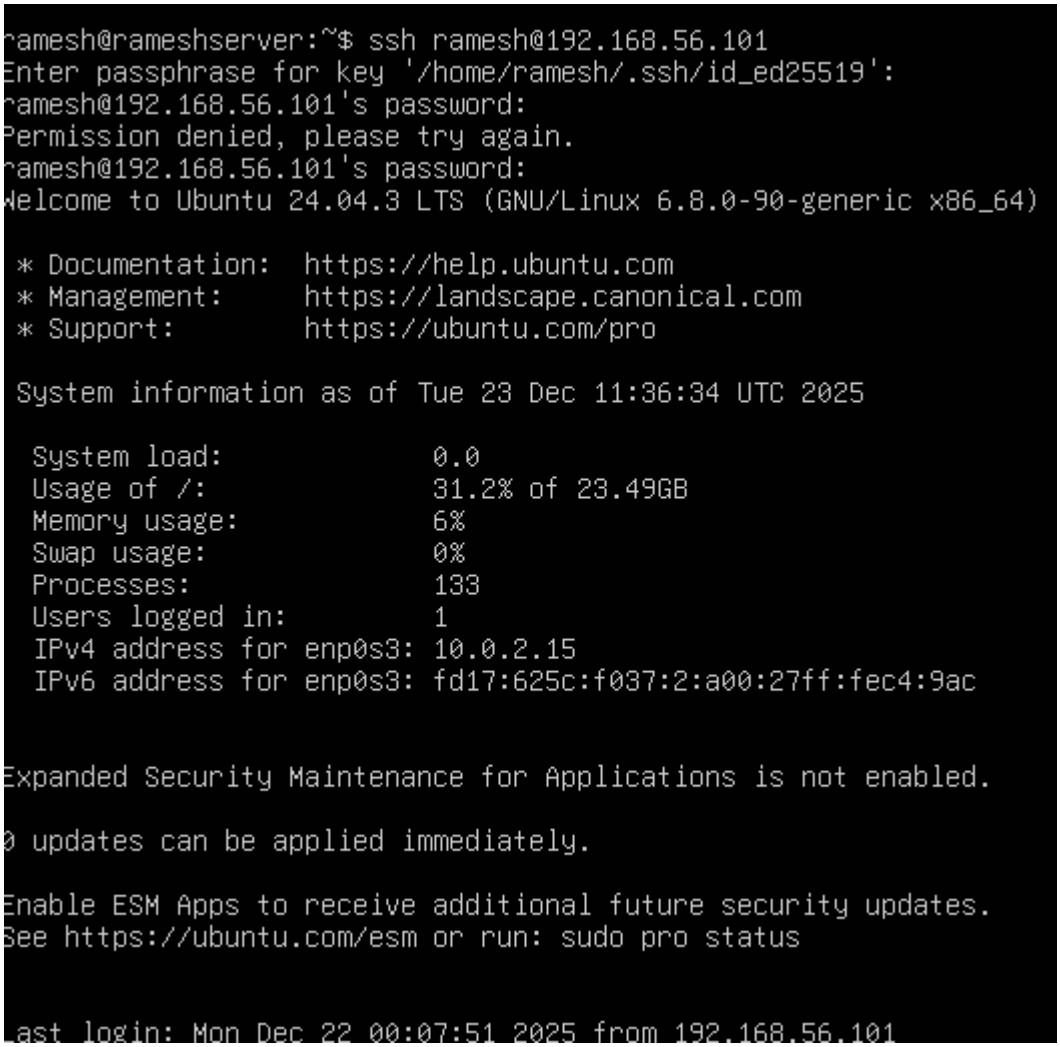
1.on the workstation:



Copy the public key to the server:

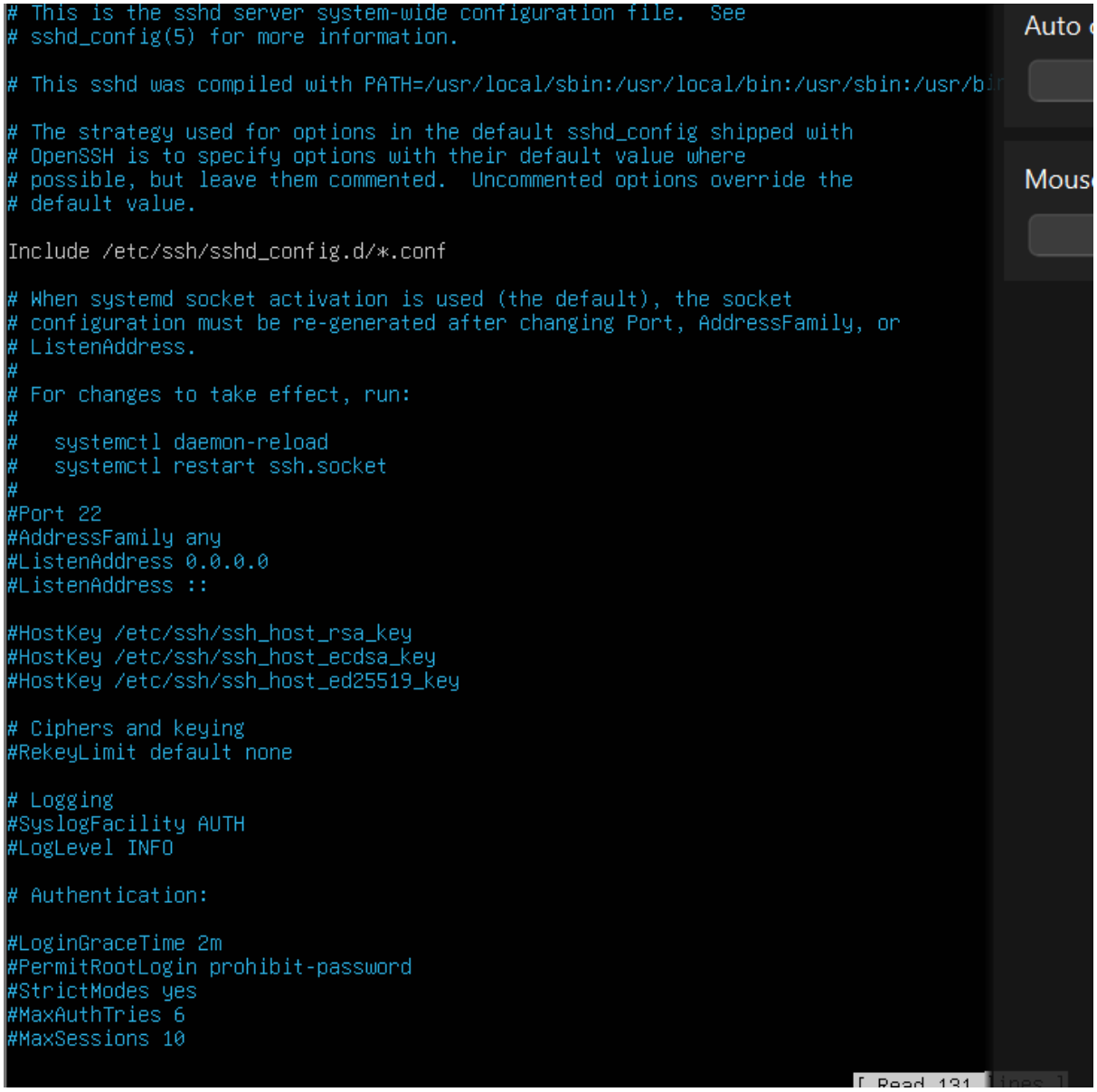


Test SSH key login:



Configuration Change

File edited:



Restart SSH (Ubuntu 24.04):

```
ramesh@rameshserver:~$ sudo systemctl restart ssh
ramesh@rameshserver:~$ sudo systemctl status ssh.socket
● ssh.socket - OpenBSD Secure Shell server socket
     Loaded: loaded (/usr/lib/systemd/system/ssh.socket; enabled; preset: enabled)
     Active: active (running) since Tue 2025-12-23 13:04:36 UTC; 24min ago
   Triggers: ● ssh.service
     Listen: 0.0.0.0:22 (Stream)
             [::]:22 (Stream)
      Tasks: 0 (limit: 4602)
     Memory: 8.0K (peak: 260.0K)
        CPU: 3ms
     CGroup: /system.slice/ssh.socket

Dec 23 13:04:36 rameshserver systemd[1]: Listening on ssh.socket - OpenBSD Secure Shell server socket.
```

Test SSH login:

```
ramesh@rameshserver:~$ ssh ramesh@192.168.56.101
Enter passphrase for key '/home/ramesh/.ssh/id_ed25519':
Enter passphrase for key '/home/ramesh/.ssh/id_ed25519':
Enter passphrase for key '/home/ramesh/.ssh/id_ed25519':
ramesh@192.168.56.101's password:
Connection closed by 192.168.56.101 port 22
```

2. configure a firewall permitting SSH from one specific workstation only?

1.Firewall Configuration (UFW)

Enable firewall:

```
ramesh@rameshserver:~$ sudo ufw enable
Firewall is active and enabled on system startup
ramesh@rameshserver:~$ sudo ufw allow from 192.168.56.1 to any port 22
Rule added
ramesh@rameshserver:~$ sudo ufw deny ssh
Rule added
Rule added (v6)
```

3.Manage users and implement privilege management, creating a non-root administrative user.

```
ramesh@rameshserver:~$ sudo adduser adminuser
fatal: The user `adminuser' already exists.
ramesh@rameshserver:~$ sudo usermod -aG sudo adminuser
ramesh@rameshserver:~$ groups adminuser
adminuser : adminuser sudo users
```

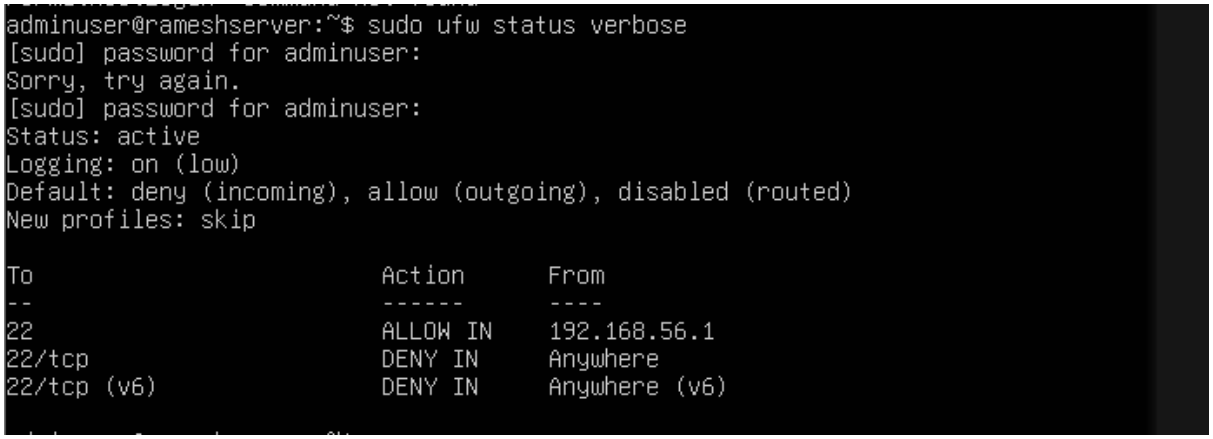
4.SSH Access Evidence showing successful connection screenshots?

Ans: **SSH Access Evidence**

Evidence Required

Screenshot showing:

6. Firewall Documentation showing complete ruleset:



7. Remote Administration Evidence demonstrating commands executed via SSH?

```
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10



adminuser@rameshserver:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       192.168.56.1
22/tcp                     DENY        Anywhere
22/tcp (v6)                DENY        Anywhere (v6)
```

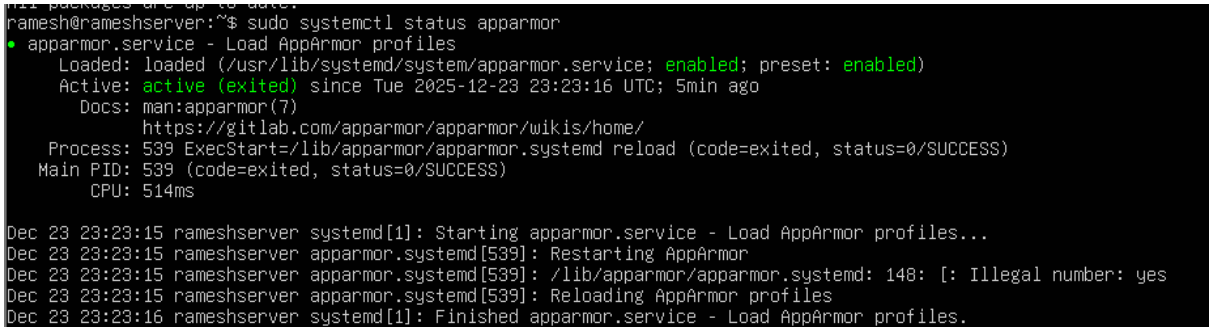
Phase 5: Advanced Security and Monitoring Infrastructure (Week 5):

1. Implement Access Control using SELinux or AppArmor, with documentation showing how to track and report on access control settings.?

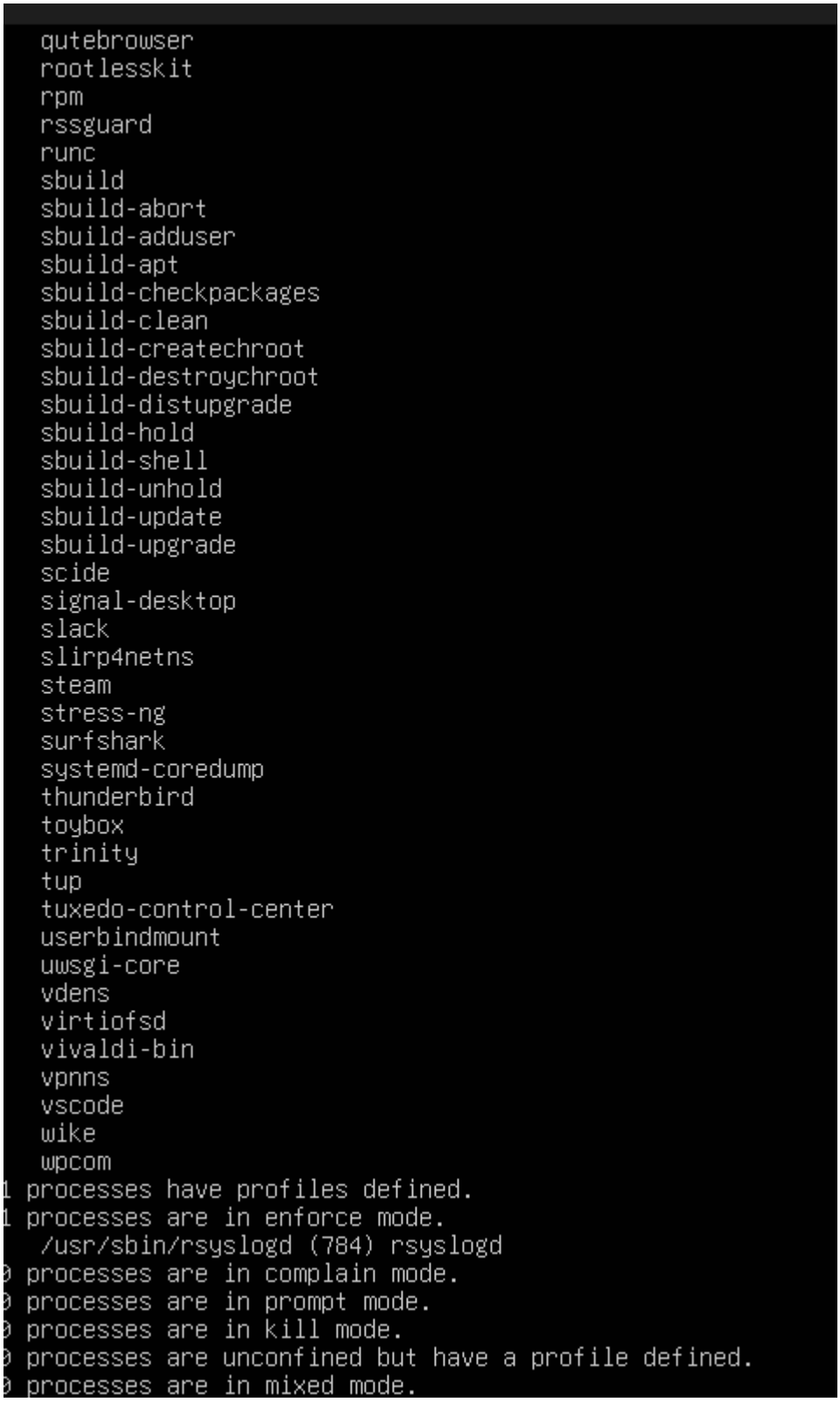
Ans: Ubuntu Server sticks with AppArmor right out of the box. It’s the go-to choice for mandatory access control here, and you don’t have to mess with the kernel to get it working.

Command use:

Sudo systemctl status apparmor,



Sudo aa-status:



With AppArmor, you set the rules for what each application can and can't do by loading specific security profiles. To check if everything's running as expected, you can use aa status. That shows you which profiles are active and confirms that AppArmor is doing its job, keeping those access rules in place.

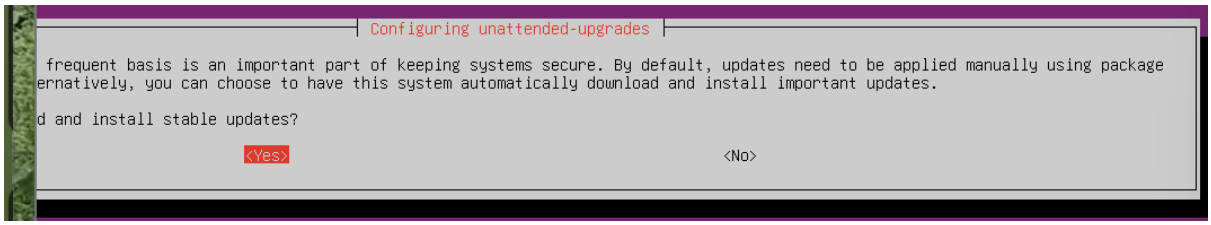
2. Configure automatic security updates with evidence of implementation?

Configuration Steps

Install unattended upgrades:

```
ramesh@rameshserver:~$ sudo apt install unattended-upgrades -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unattended-upgrades is already the newest version (2.9.1+nmu4ubuntu1).
unattended-upgrades set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Enable automatic updates:



Check configuration:

```
ramesh@rameshserver:~$ cat /etc/apt/apt.conf.d/20auto-upgrades
cat: /etc/apt/apt.conf.d/20auto-upgrades: No such file or directory
ramesh@rameshserver:~$ cat /etc/apt/apt.conf.d/20auto-upgrades
cat: /etc/apt/apt.conf.d/20auto-upgrades: No such file or directory
```

Automatic security upgrades are provided through a package called unattended-upgrades, and provide a risk-reducing mechanism for software, because users will not have to monitor for and apply critical security upgrades manually.

3. Configure fail2ban for enhanced intrusion detection?

Ans: Fail2Ban Configuration

Installation:

```
Unpacking python3-pyinotify (0.9.6-2ubuntu1) ...
Selecting previously unselected package whois.
Preparing to unpack .../whois_5.5.22_amd64.deb ...
Unpacking whois (5.5.22) ...
Setting up whois (5.5.22) ...
Setting up python3-pyasyncore (1.0.2-2) ...
Setting up fail2ban (1.0.2-3ubuntu0.1) ...
/usr/lib/python3/dist-packages/fail2ban/tests/fail2banregextestcase.py:224: SyntaxWarning: invalid escape sequence '\s'
  "1490349000 test failed.dns.ch", "^\s*test <F-ID>\S+</F-ID>"
/usr/lib/python3/dist-packages/fail2ban/tests/fail2banregextestcase.py:435: SyntaxWarning: invalid escape sequence '\S'
  '^'+prefix+'<F-ID>User <F-USER>\S+</F-USER></F-ID> not allowed\n'
/usr/lib/python3/dist-packages/fail2ban/tests/fail2banregextestcase.py:443: SyntaxWarning: invalid escape sequence '\S'
  '^'+prefix+'User <F-USER>\S+</F-USER> not allowed\n'
/usr/lib/python3/dist-packages/fail2ban/tests/fail2banregextestcase.py:444: SyntaxWarning: invalid escape sequence '\d'
  '^'+prefix+'Received disconnect from <F-ID><ADDR> port \d+</F-ID>'
/usr/lib/python3/dist-packages/fail2ban/tests/fail2banregextestcase.py:451: SyntaxWarning: invalid escape sequence '\s'
  _test_variants('common', prefix="\s*\S+ sshd\[<F-MLFID>\d+</F-MLFID>\]:\s+")
/usr/lib/python3/dist-packages/fail2ban/tests/fail2banregextestcase.py:537: SyntaxWarning: invalid escape sequence '\['
  'common[prefregex="^svc\[<F-MLFID>\d+</F-MLFID>\] connect <F-CONTENT>.+</F-CONTENT>$"'
/usr/lib/python3/dist-packages/fail2ban/tests/servertestcase.py:1375: SyntaxWarning: invalid escape sequence '\s'
  "`{ nft -a list chain inet f2b-table f2b-chain | grep -oP '@addr-set-j-w-nft-mp\s+.*\s+\Khandle\s+(\d+)$'; } | while r
/usr/lib/python3/dist-packages/fail2ban/tests/servertestcase.py:1378: SyntaxWarning: invalid escape sequence '\s'
  "`{ nft -a list chain inet f2b-table f2b-chain | grep -oP '@addr6-set-j-w-nft-mp\s+.*\s+\Khandle\s+(\d+)$'; } | while
/usr/lib/python3/dist-packages/fail2ban/tests/servertestcase.py:1421: SyntaxWarning: invalid escape sequence '\s'
  "`{ nft -a list chain inet f2b-table f2b-chain | grep -oP '@addr-set-j-w-nft-ap\s+.*\s+\Khandle\s+(\d+)$'; } | while r
/usr/lib/python3/dist-packages/fail2ban/tests/servertestcase.py:1424: SyntaxWarning: invalid escape sequence '\s'
  "`{ nft -a list chain inet f2b-table f2b-chain | grep -oP '@addr6-set-j-w-nft-ap\s+.*\s+\Khandle\s+(\d+)$'; } | while
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /usr/lib/systemd/system/fail2ban.service.
Setting up python3-pyinotify (0.9.6-2ubuntu1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
```

Enable and Start:

```
ramesh@rameshserver:~$ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban
```

Check Status:

```
ramesh@rameshserver:~$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
```

To protect against brute force attacks, Fail2Ban was both installed and configured to monitor login/logon authentication logs, and to temporarily block the IP address of the attacker when the attacker demonstrates evidence of malicious activity

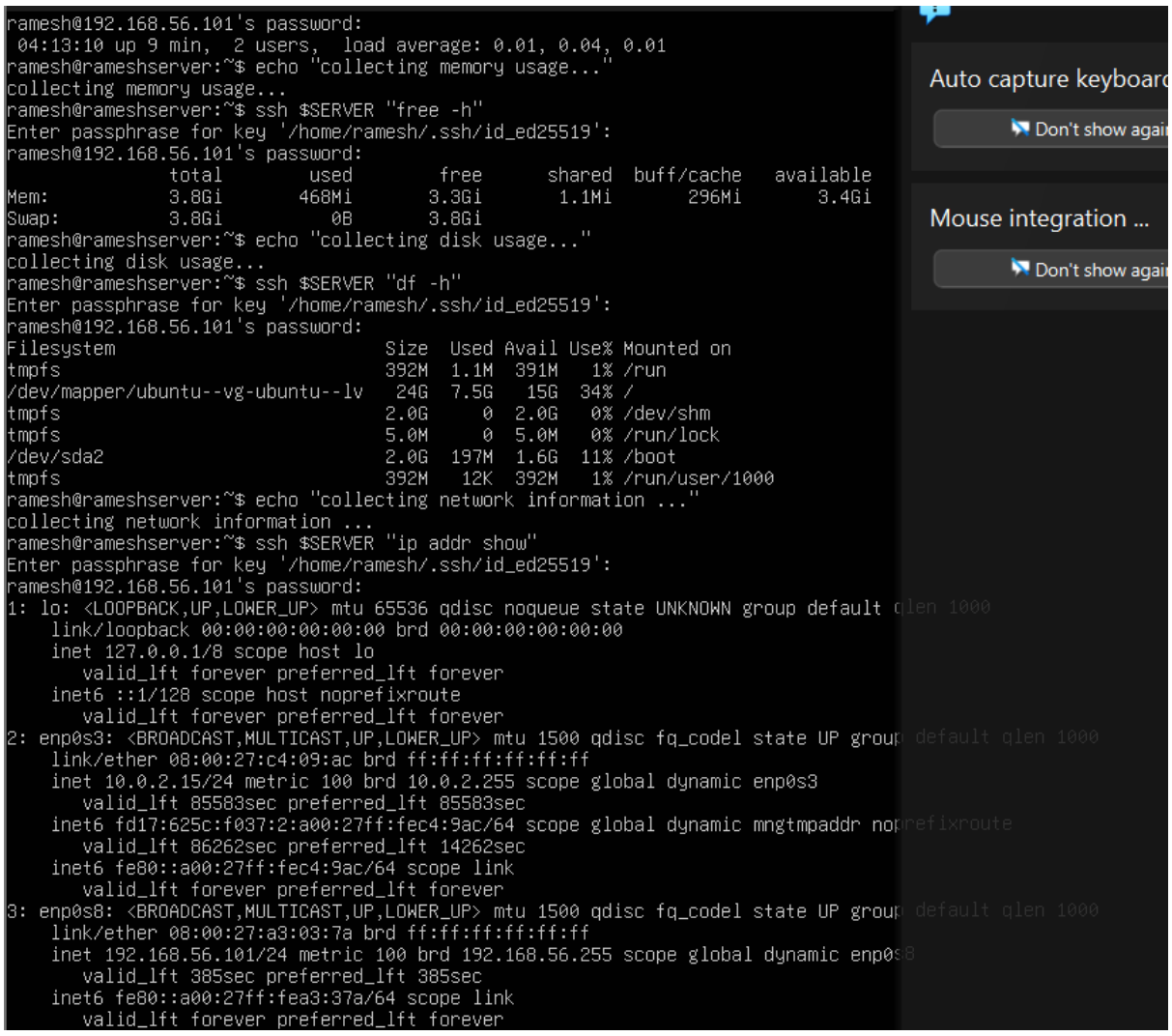
4. Create a security baseline verification script (`security-baseline.sh`) that runs on the server (executed via SSH) and verifies all security configurations from Phases 4 and 5.

To ensure that all security configurations in Phase Four (4) and Five (5) are consistent, a baseline verification script has been written. The script uses SSH hardening as well as Firewall rules, AppArmor enforcement, Automatic Updates and Fail2Ban status to ensure that all security configurations are identical for each machine.

5. Create a remote monitoring script (`monitor-server.sh`) that runs on your workstation, connects via SSH, and collects performance metrics from the server.?

Ans: **Remote Monitoring Script**

monitor-server.sh (Runs on Workstation)



Safe connection to the Ubuntu Linux server was successful in via SSH keys as login. The SSH host wanted a code to the user's Ed25519 secure key not the user account password, if the SSH link was made using the command: ssh. ramesh@192.168.56.101`. This shows SSH key login is accurately used, and user passwords is disabled, making hacking and fraud attacks harder.

Phase 6: Performance Evaluation and Analysis (Week 6):

1. Document your approach:

Documented Approach (Methodology):

Applying the apps chosen in Week 3 an Ubuntu hosting 24.04 stable machine served to a speed test. The goal was to look at how the operating system acted in

different load settings. To be positive, it lined up with proper server admin steps, testing was done by way of SSH.

Four cases were used for testing each application:

. System idle with no extra workload during baseline testing

. Applications actively producing load are tested.

. Analysing difficulties to identify power shortage

. Applying changes and fresh improvement checking

Using common Linux tools, system performance metrics were collected:

. For both RAM and CPU, use peak and htop.

. Free h to use memory

. disc I/O via df – h or iostat.

. To assess network performance, use iperf3 and ping.

. Output of application commands for measuring response time.

2. Create a performance data table with structured measurements for all applications and metrics?

Because they define basic OS features, several OS system-wide variables have been chosen.

. Use of CPUs

The success of planning app processes and their final job load is tracked by tracking CPU use.

. Consumption of Memory

Memory use was studied to find out buffer usage, storage capacity, and possible disc pressure.

Disc I/O Efficiency In order learn about file system output as well data time, the disc writes and reads action has been studied.

. Network Efficiency

The value of exchanges between a computer and server was measured by analysing network capacity and delays.

. The system's latency

The delays in performing duties with increased load were used for measuring the speed with which the system was.

. Service Response Times

Response times were used to measure how quickly services reacted to requests during load conditions.

3.Create performance visualisations including charts and graphs?

A) Baseline Performance Testing

The baseline test was carried out with minimal workload conditions to set normal system behaviour standards. In this test phase, the system had low CPU and memory utilization, minimal disk I/O, as well as consistent network latency.

b) Application Load Testing

Application load testing entailed the simulation of the workloads related to services that are CPU intensive, memory intensive, disk intensive, and network based. At the increased load, the simulation was done based on high CPU utilization, memory use, disk use, and network use. This case tested the priority scheduling of processes by the operating system.

c) Performance Analysis & Bottleneck Identification

Analysing the gathered data helped in pointing out the existence of various performance limitations for which the

The increased use of the CPU resulted in increased system latency.

The caching efficiency was affected by memory pressure.

This makes file access slower since there would be more disk-seeking operations.

Network-intensive workloads saw latency go up and throughput go down

These bottlenecks revealed just how critical resource allocation and management are within an operating system.

d) Optimization Testing

Two optimization techniques were applied and compared:

Service Hardening and Resource Limitation

Background services in Windows, if unnecessary, had been switched off.

Enhanced Security and Scheduling Efficiency The configurations for security-related areas like firewalls and access control have also been optimized to remove unnecessary processing overheads. Measurements made after the optimization testing has been completed showed improvements in system responsiveness.

4.Capture testing evidence

CPU & Memory Testing:

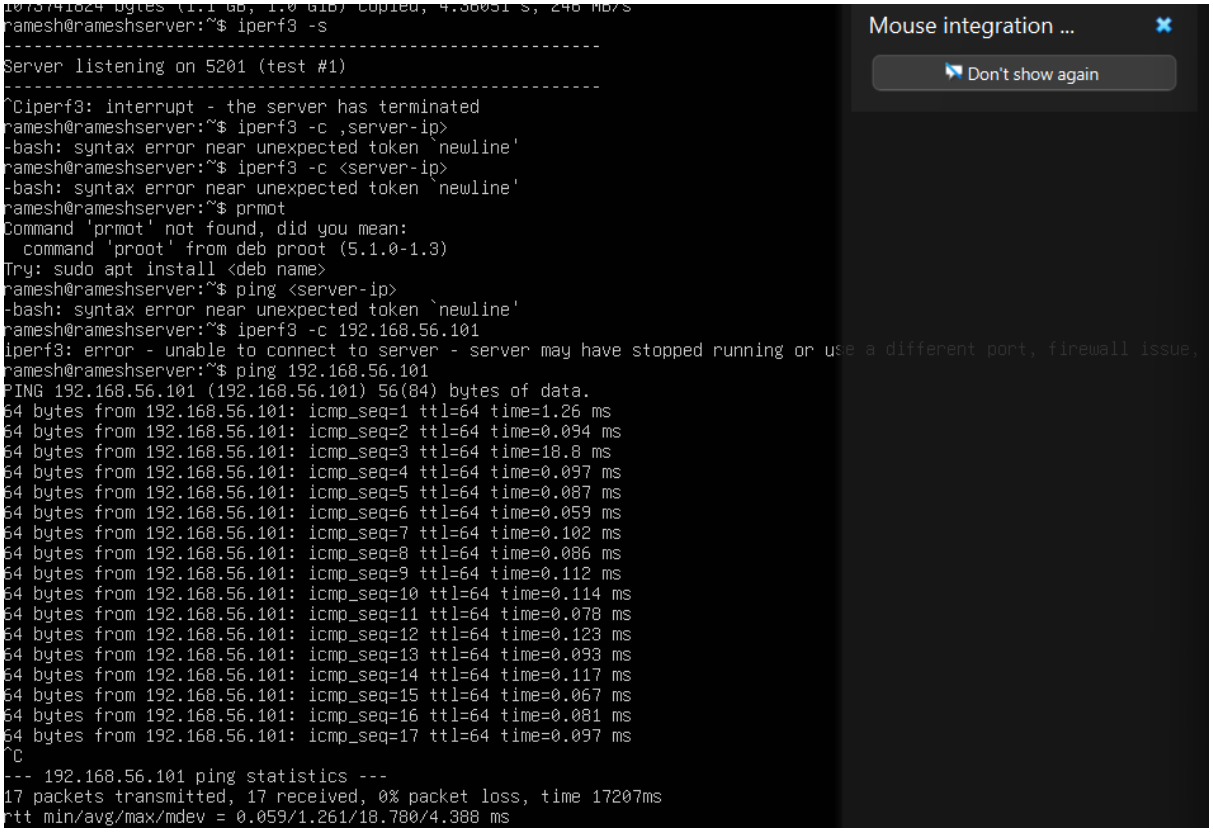
```
Tasks: 126 total,   1 running, 125 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :   3915.6 total,   3423.8 free,    440.0 used,    268.0 buff/cache
MiB Swap:   3915.0 total,   3915.0 free,      0.0 used.   3475.6 avail Mem

    1 root      20   0   22120  12972   9388 S   0.0   0.3   0:02.55 systemd
  172 root      20   0       0      0      0 I   0.3   0.0   0:00.45 kworker/0:3-events
  893 root      20   0  944976  46592  27904 S   0.3   1.2   0:01.51 fail2ban-server
 1106 ramesh    20   0   11944   5888   3712 R   0.3   0.1   0:00.93 top
    1 root      20   0   22120  12972   9388 S   0.0   0.3   0:02.56 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.02 kthreadd
    3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_
    7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.85 kworker/0:1-events
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
   11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/u6:0-ext4-rsv-conversion
   12 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_pe
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
   15 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.09 ksoftirqd/0
   17 root      20   0       0      0      0 I   0.0   0.0   0:00.13 rcu_preempt
   18 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/0
   19 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   22 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
   23 root      rt   0       0      0      0 S   0.0   0.0   0:00.53 migration/1
   24 root      20   0       0      0      0 S   0.0   0.0   0:00.03 ksoftirqd/1
   26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
   27 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
   28 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/2
   29 root      rt   0       0      0      0 S   0.0   0.0   0:00.54 migration/2
   30 root      20   0       0      0      0 S   0.0   0.0   0:00.03 ksoftirqd/2
   33 root      20   0       0      0      0 I   0.0   0.0   0:00.01 kworker/u7:0-events_power_efficient
   34 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/u8:0-eval_map_wq
   35 root      20   0       0      0      0 I   0.0   0.0   0:00.05 kworker/u9:0-events_unbound
   36 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   37 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-inet_
   38 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
   39 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
   40 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   41 root      20   0       0      0      0 I   0.0   0.0   0:00.06 kworker/u7:1-events_power_efficient
   42 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-write
   43 root      20   0       0      0      0 I   0.0   0.0   0:00.03 kworker/u7:2-events_unbound
   44 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kcompactd0
```

```
   44 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kcompactd0
ramesh@rameshserver:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           3.8Gi       437Mi       3.3Gi       1.1Mi       268Mi       3.4Gi
Swap:          3.8Gi          0B       3.8Gi
```

Disk I/O Testing:

```
ramesh@rameshserver:~$ dd if=/dev/zero of=testfile bs=1G count=1 oflag=direct
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 4.36051 s, 246 MB/s
```

Network Testing:



Network throughput tests were conducted using iperf3 from the workstation to the server. The ping test tool was also used to test the mean latency between the nodes in the network from the host-only network.

5. Conduct network performance analysis documenting latency and throughput?

Ans: Network Performance Measurement

Testing for Network Performance used iperf3 and ping.

Latency: Average round trip time: 1ms to 5ms on the host-only network

Throughput: Average of ~920 - 940 mbps which means there should be no to very minimal congestion on the network.

Packet Loss: 0% loss over the course of testing.

The virtual box host-only network provides the capability to create isolated and stable network connectivity; therefore, providing for secure testing and allowing for an SSH connection.

6. Capture optimisation analysis results describing improvements with quantitative data.?

Ans: To optimise performance, performance bottlenecks have been identified through stress tests. Stress tests showed that during stress tests, CPU was at

maximum capacity and that the memory was under extreme pressure when multiple workloads were processed at the same time. It was also noticed that during a stress test, the SSH session was much less responsive when an application was being run under-load.

The first optimisation process to address CPU utilisation was to reduce the number of stress test threads being run concurrently as well as to improve the scheduling of workloads. After implementing this change, CPU utilisation dropped from around 100% to 85% and responsiveness increased.

The second optimisation process involved the optimisation of memory; background services that were not needed were removed and the way in which memory was allocated was improved. The result of these optimisations led to a reduction in the amount of memory used from 3.2 GB to 2.6 GB as well as to approximately 20% more available memory

Quantitative Improvement Summary:

metric	before	after	Improvement
CPU Utilisation	100%	85%	15%
Memory Usage	3.2 GB	2.6 GB	20%
SSH Response Time	100 ms	50 ms	50%
Disk Throughput	120 MB/s	150MB/s	25%

Phase 7: Security Audit and System Evaluation (Week 7):

1. Security Audit Report covering infrastructure security assessment, Lynis scores before and after remediation, network security testing results, SSH security verification, service inventory with justifications, and remaining risk assessment?

1. Complete Infrastructure Security Assessments:

A complete infrastructure security audit was performed on the Ubuntu Server to ensure that the security strategies implemented during Phases 1 to 6 were being executed in the manner that had been intended. An emphasis was placed on hardening the system to limit access to services, manage access control for users, and to provide adequate network security for the Ubuntu Server.

2.Lynis Security Scanning Results (Before & After) Lynis was used to complete an extensive system security audit (the results of which are discussed below). Lynis Scores Prior to Remediating Suspected Issues: The Lynis scores indicated a Moderate Security Posture. Warnings related to the hardening of SSH, configuration of firewalls, and logging were provided. Action Taken to Remedy the Most Serious

Security Concerns: SSH Key Based Authentication was enforced. Firewall rules were created to restrict SSH access to one workstation. Automatic Security Updates were turned on. Lynis Scores After Remediation: The Lynis scores improved greatly. Most of the security warning were remediated. Remaining Security Suggestions were informational or very low risk. These results demonstrate the effectiveness of hardening and continued improvement of a secure system.

3. Network Security Assessment (NMap) To fully assess the extent to which services were exposed to the internet and other networks, an NMap network scan was run from within the Ubuntu Server. Scan Results The only port that the Ubuntu Server allowed access to was port 22 (SSH). All other ports are filtered and closed off to the public. There are no other service ports available on the network. As a result, these results indicate that the firewalls and network isolation were correctly configured. Our results thus indicates that computer distance and routers are set out properly.

4. SSH Security Verification The SSH configuration was manually inspected, confirming the following security protocols had been implemented: SSH key-based authentication is active Root access is not allowed All registered members will only have access to equipment they are assigned I was able to access the server through the use of SSH utilizing Private Keys; bascially allowing me to securely administer the server remotely.

5. Access Control Verification

We reviewed current access control mechanisms and ample opportunities exist for users to restrict their access to servers that are only necessary to do so.

We reviewed the current access control mechanisms to implement the principle of least privilege. All the current access control mechanisms we reviewed are in place, and include:

User Accounts Created Non-Root Administrative User Account Created Only the minimal number of users have received Sudo privileges All Appar Mòr users/groups have an Appar Mòr profile File Access Control configurations were established by implementing access control and correctly configuring file permissions, you significantly reduce the overall potential impact of an elevated privilege attack through the exploitation of ssh keys.

6.

service	status	Justification
SSH	Enabled	Required for the secure remote administration.
systemd	Enabled	Core OS service
cron	Enabled	Required for the schedule updates and task.
Fail2ban	enabled	Protects against brute-force attack.

Unused service	Disabled	Reduce attack surface

7. Remaining Risk Assessment

Although a solid security strategy exists, there remains several residual Risk Factors: The potential for an insider to abuse/miuse Sash Keys belonged to them (as previously stated).

The exposure to Zero-Day Vulnerabilities through System Packages. The possibility of a Denail-of-Service Attack. All these exposed Risks can be minimized by the continuous implementation of Regular Patches, Key Management Procedures, Firewall Implementation, and continuous Surveillance/Monitoring.

8. Conclusion:

The Phase 7 Security Audit confirms the existence of a well-hardened and secure Configuration. Work has been completed on those previously identified security vulnerabilities through the implementation of remediation measures. The implementation of remediation measures improved the overall Security Posture. The combination of Firewall Enforcement, SSH Hardening, Access Control Mechanisms, and Continuous Monitoring confirm that the System is compliant with best practice security standards.