# Week 2: Security Planning and Performance Testing Methodology

### 1. Aim of the Week

The aim of Week 2 was to plan a secure baseline configuration for the Linux system and to design a performance testing methodology. This week focused on understanding what security controls are required, why they are important, and how system performance can be monitored remotely before any security configurations are applied.

No security settings were implemented during this week. Instead, the focus was on planning and preparation for later phases of the coursework.

### 2. Performance Testing Plan

Performance testing is important to understand how system resources such as CPU, memory, disk, and network are used during normal operation and under load. Monitoring will be carried out remotely from the workstation using command-line tools over SSH, ensuring the server remains headless and reflects real-world administration practices.

The initial performance metrics will be collected as a baseline before any workloads are applied. These measurements will later be compared with results collected during application testing.

**Planned Monitoring Tools**

| Resource | Command | Purpose |
|---|---|---|
| Memory usage | free -h | Display RAM usage and availability |
| Disk usage | df -h | Show disk space usage |
| CPU activity | top | Monitor processor utilisation |
| Network status | ip addr, ping | Check connectivity and latency |

The screenshots taken using free -h and df -h demonstrate how memory and disk usage can be observed directly from the command line. These commands provide a clear overview of available resources and will be used throughout the performance testing phase.

### 3. Security Configuration Checklist

A security baseline is required to protect the system from common threats. The following checklist outlines the security configurations that will be implemented in later phases of the coursework.

| Security Area | Planned Configuration | Reason |
|---|---|---|
| SSH Hardening | Disable password login and use SSH keys | Prevent brute-force attacks |
| Firewall | Allow SSH only from the workstation IP | Restrict unauthorized access |
| Mandatory Access Control | Enable AppArmor or SELinux | Limit process permissions |
| Automatic Updates | Enable unattended security updates | Patch vulnerabilities automatically |
| User Privileges | Use non-root administrative user | Reduce risk of full system compromise |
| Network Security | Close unused ports | Minimise attack surface |

This checklist ensures that the system follows best security practices before advanced monitoring and performance testing is conducted.

### 4. Threat Model

A threat model was created to identify potential security risks and define mitigation strategies.

**Threat 1: Brute-force SSH attacks**

- **Description:** Attackers may attempt to guess SSH credentials.
- **Impact:** Unauthorized access to the system.
- **Mitigation:** Disable password authentication and enforce key-based SSH access.

**Threat 2: Unpatched software vulnerabilities**

- **Description:** Outdated software may contain known security flaws.
- **Impact:** Attackers could exploit vulnerabilities to gain access.
- **Mitigation:** Enable automatic security updates to ensure patches are applied promptly.

**Threat 3: Privilege escalation**

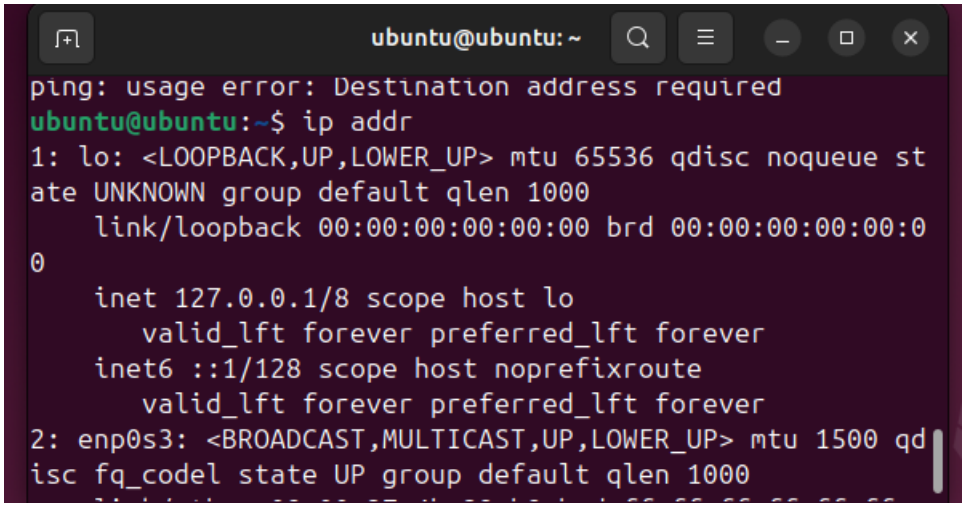
- **Description:** A compromised user could gain elevated privileges.
- **Impact:** Full system compromise.
- **Mitigation:** Apply least-privilege principles and restrict root access.

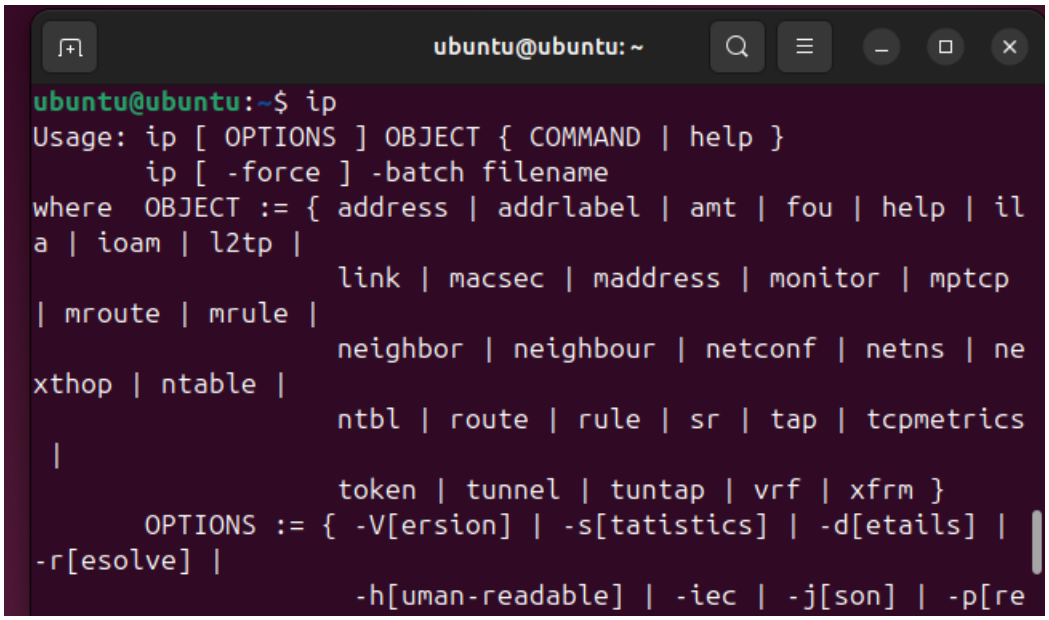
Identifying these threats early helps guide the security controls that will be implemented in later weeks.

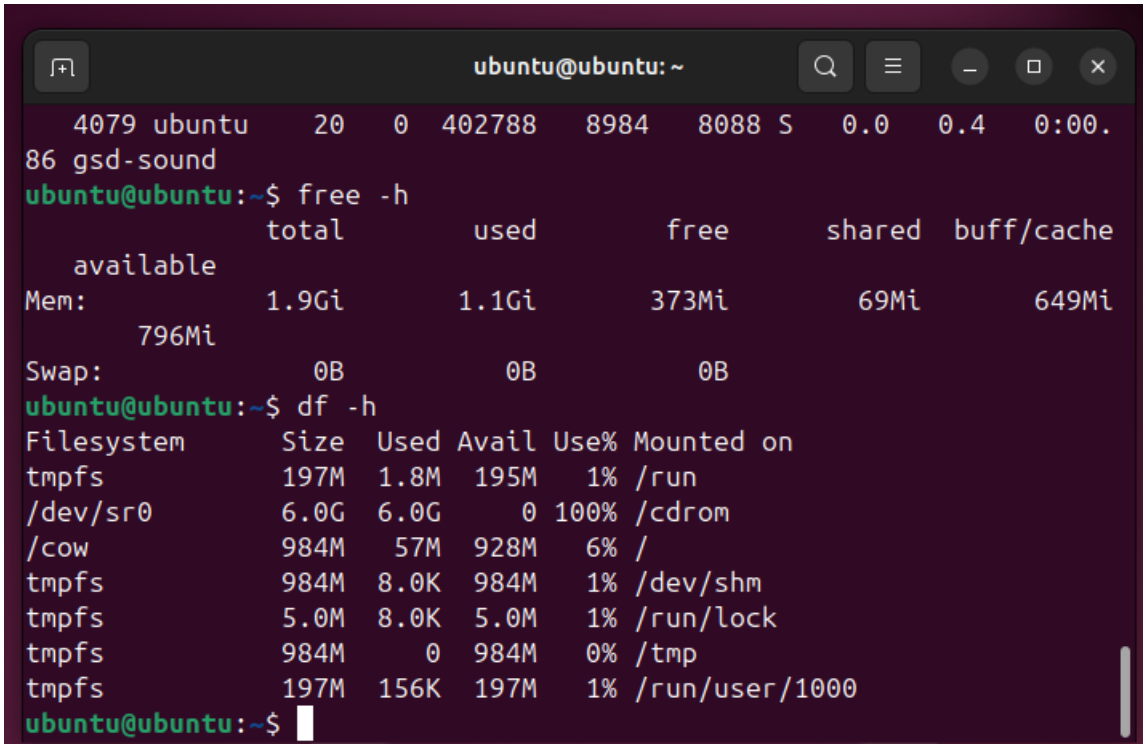
## 5. Reflection

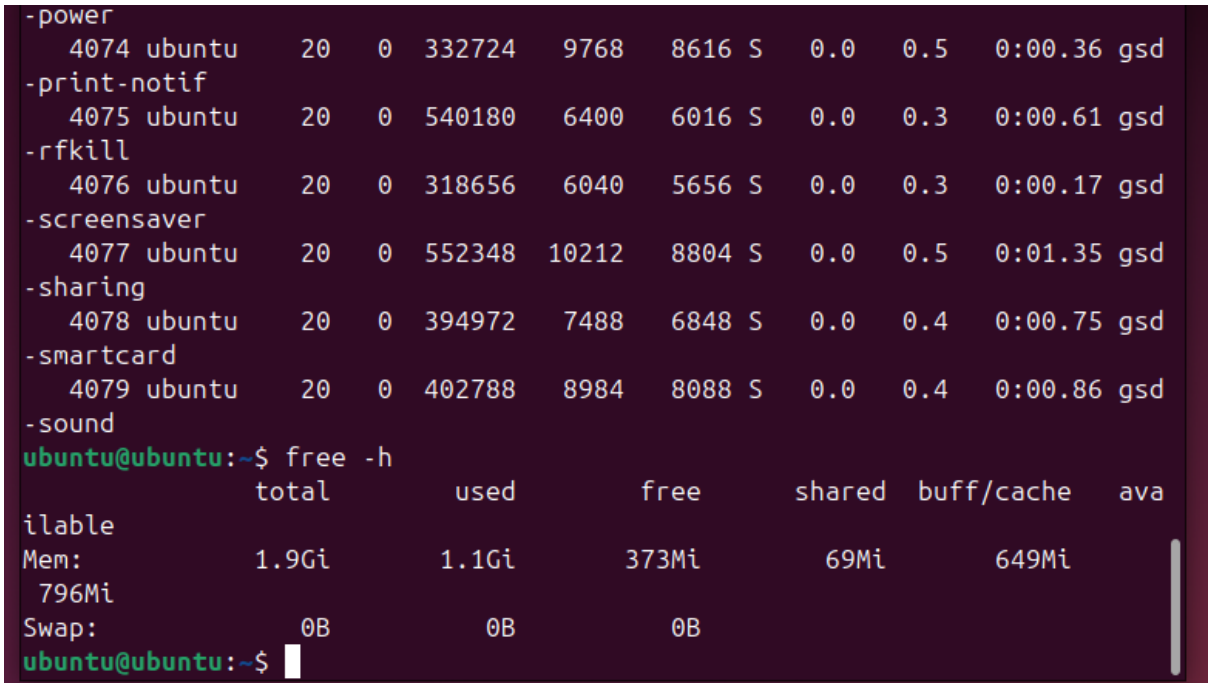
This week helped me understand the importance of planning before applying security configurations. Designing a performance testing methodology and security baseline clarified how operating system security and performance are closely related. The use of commands such as free -h and df -h improved my understanding of how system resources can be monitored using the command line.

This planning phase has prepared me for the next stages of the coursework, where these security measures and monitoring techniques will be implemented and tested in practice.


ubuntu@ubuntu: ~
ping: usage error: Destination address required
ubuntu@ubuntu:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue st
ate UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:0
0
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
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qd
isc fq_codel state UP group default qlen 1000


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
ip [ -force ] -batch filename
where OBJECT := { address | addrlabel | amt | fou | help | il
a | ioam | l2tp |
link | macsec | maddress | monitor | mptcp
| mroute | mrule |
neighbor | neighbour | netconf | netns | ne
xthop | ntable |
ntbl | route | rule | sr | tap | tcpmetrics
|
token | tunnel | tuntap | vrf | xfrm }
OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] |
-r[esolve] |
-h[uman-readable] | -iec | -j[son] | -p[re


ubuntu@ubuntu: ~
4079 ubuntu 20 0 402788 8984 8088 S 0.0 0.4 0:00.86 gsd-sound
ubuntu@ubuntu:~$ free -h
total used free shared buff/cache available
Mem: 1.9Gi 1.1Gi 373Mi 69Mi 649Mi 796Mi
Swap: 0B 0B 0B
ubuntu@ubuntu:~$ df -h
Filesystem Size Used Avail Use% Mounted on
tmpfs 197M 1.8M 195M 1% /run
/dev/sr0 6.0G 6.0G 0 100% /cdrom
/cow 984M 57M 928M 6% /
tmpfs 984M 8.0K 984M 1% /dev/shm
tmpfs 5.0M 8.0K 5.0M 1% /run/lock
tmpfs 984M 0 984M 0% /tmp
tmpfs 197M 156K 197M 1% /run/user/1000
ubuntu@ubuntu:~$


-power
4074 ubuntu 20 0 332724 9768 8616 S 0.0 0.5 0:00.36 gsd-print-notif
4075 ubuntu 20 0 540180 6400 6016 S 0.0 0.3 0:00.61 gsd-rfkill
4076 ubuntu 20 0 318656 6040 5656 S 0.0 0.3 0:00.17 gsd-screensaver
4077 ubuntu 20 0 552348 10212 8804 S 0.0 0.5 0:01.35 gsd-sharing
4078 ubuntu 20 0 394972 7488 6848 S 0.0 0.4 0:00.75 gsd-smartcard
4079 ubuntu 20 0 402788 8984 8088 S 0.0 0.4 0:00.86 gsd-sound
ubuntu@ubuntu:~$ free -h
total used free shared buff/cache available
Mem: 1.9Gi 1.1Gi 373Mi 69Mi 649Mi 796Mi
Swap: 0B 0B 0B
ubuntu@ubuntu:~$