

# Week 6 — Performance Evaluation & Analysis

---

## Overview

Week 6 focuses on evaluating operating system behaviour under a range of controlled workloads using quantitative performance testing. The aim is to analyse how the Linux operating system manages CPU scheduling, memory allocation, disk I/O, and network communication under both idle and stressed conditions.

All performance monitoring is conducted remotely via SSH using command-line tools and automated scripts developed in Week 5, reflecting professional system administration practices. The results are used to identify performance bottlenecks, evaluate optimisation strategies, and critically analyse trade-offs between system performance, stability, and security.

---

## Objectives

- Execute baseline and load-based performance testing
  - Measure CPU, memory, disk I/O, and network performance
  - Identify system bottlenecks under stress
  - Apply and evaluate performance optimisations
  - Analyse operating system trade-offs using quantitative data
- 

## Deliverables

- Performance testing methodology
  - Structured performance data tables
  - Performance graphs and visualisations
  - Network latency and throughput analysis
  - Optimisation results with before/after comparison
- 

## 1. Testing Methodology

### 1.1 Baseline Testing

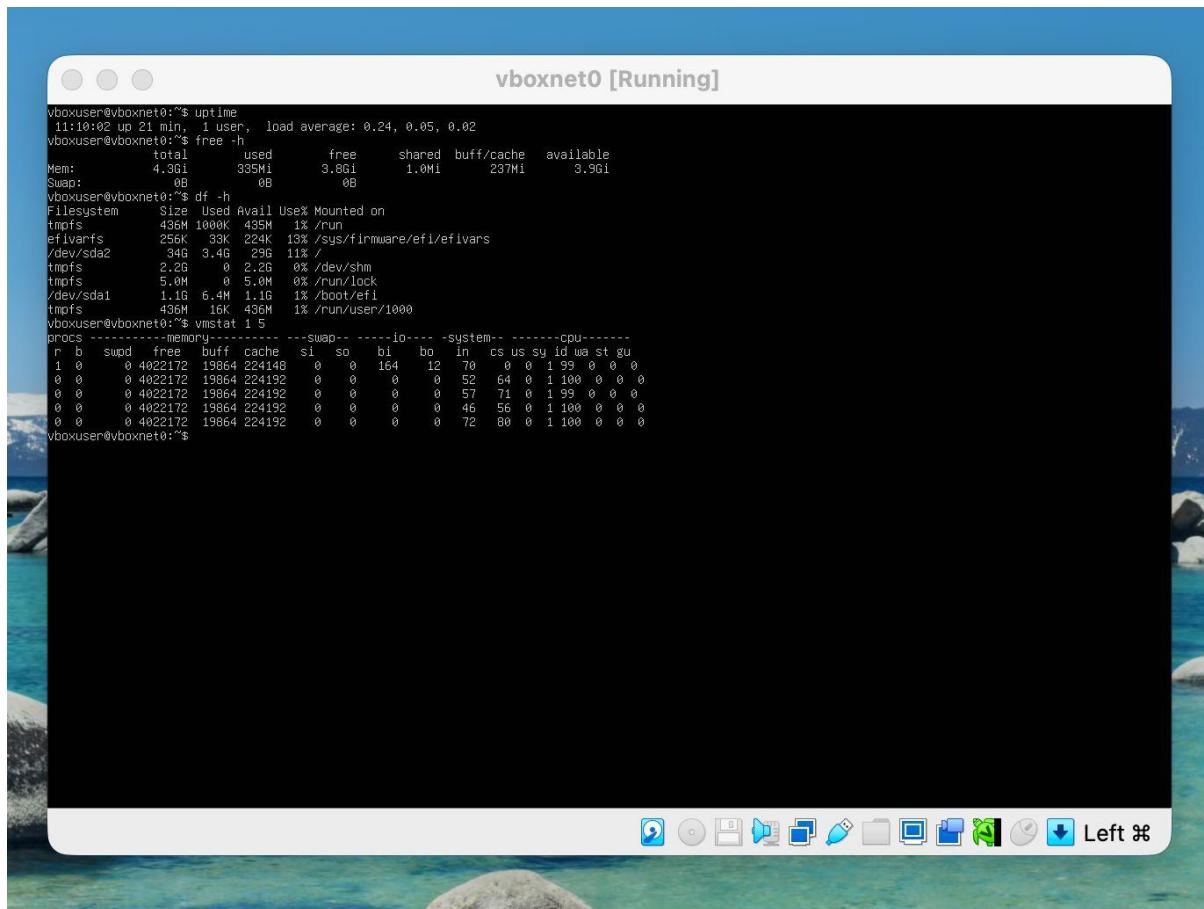
Baseline performance testing was conducted immediately after system boot with no additional workloads running. This establishes a reference point against which all subsequent performance results are compared.

### Commands Used:

```
uptime  
free -h  
df -h  
vmstat 1 5
```

### Metrics Collected:

- CPU idle percentage and load averages
- Available and used memory
- Disk utilisation
- System responsiveness



The screenshot shows a terminal window titled "vboxnet0 [Running]" with a blue header bar. The terminal displays the output of several commands:

```
vboxuser@vboxnet0:~$ uptime  
11:10:02 up 21 min, 1 user, load average: 0.24, 0.05, 0.02  
vboxuser@vboxnet0:~$ free -h  
total used free shared buff/cache available  
Mem: 4.3Gi 335Mi 3.8Gi 1.0Mi 237Mi 3.9Gi  
Swap: 0B 0B 0B  
vboxuser@vboxnet0:~$ df -h  
Filesystem Size Used Avail Use% Mounted on  
tmpfs 436M 1000K 435M 1% /run  
efivarsfs 256K 33K 224K 13% /sys/firmware/efi/efivars  
/dev/sda2 34G 3.4G 29G 11% /  
tmpfs 2.2G 0 2.2G 0% /dev/shm  
tmpfs 5.0M 0 5.0M 0% /run/lock  
/dev/sdai 1.1G 6.4M 1.1G 1% /boot/efi  
tmpfs 436M 16K 436M 1% /run/user/1000  
vboxuser@vboxnet0:~$ vmstat 1 5  
procs --memory-- --swap-- --io-- --system-- --cpu--  
r b smpd free buff cache si so bi bo in cs us sy id wa st gu  
1 0 0 4022172 19864 224148 0 0 164 12 70 0 0 1 99 0 0 0  
0 0 0 4022172 19864 224192 0 0 0 0 52 64 0 1 100 0 0 0  
0 0 0 4022172 19864 224192 0 0 0 0 57 71 0 1 99 0 0 0  
0 0 0 4022172 19864 224192 0 0 0 0 46 56 0 1 100 0 0 0  
0 0 0 4022172 19864 224192 0 0 0 0 72 80 0 1 100 0 0 0  
vboxuser@vboxnet0:~$
```

The terminal window has a dark background and light-colored text. At the bottom, there is a standard Linux-style toolbar with icons for file operations, terminal functions, and system status.

**Figure W6-1:** Baseline system performance metrics collected remotely via SSH.

## 1.2 Load Testing Scenarios

Each workload selected in Week 3 was tested individually to isolate its impact on system resources.

## Tested Workloads:

- CPU-intensive: stress-ng
- Memory-intensive: stress-ng
- Disk I/O-intensive: dd
- Network-intensive: iperf3
- Server workload: nginx

## Example Command:

```
stress-ng --cpu 2 --timeout 120s
```

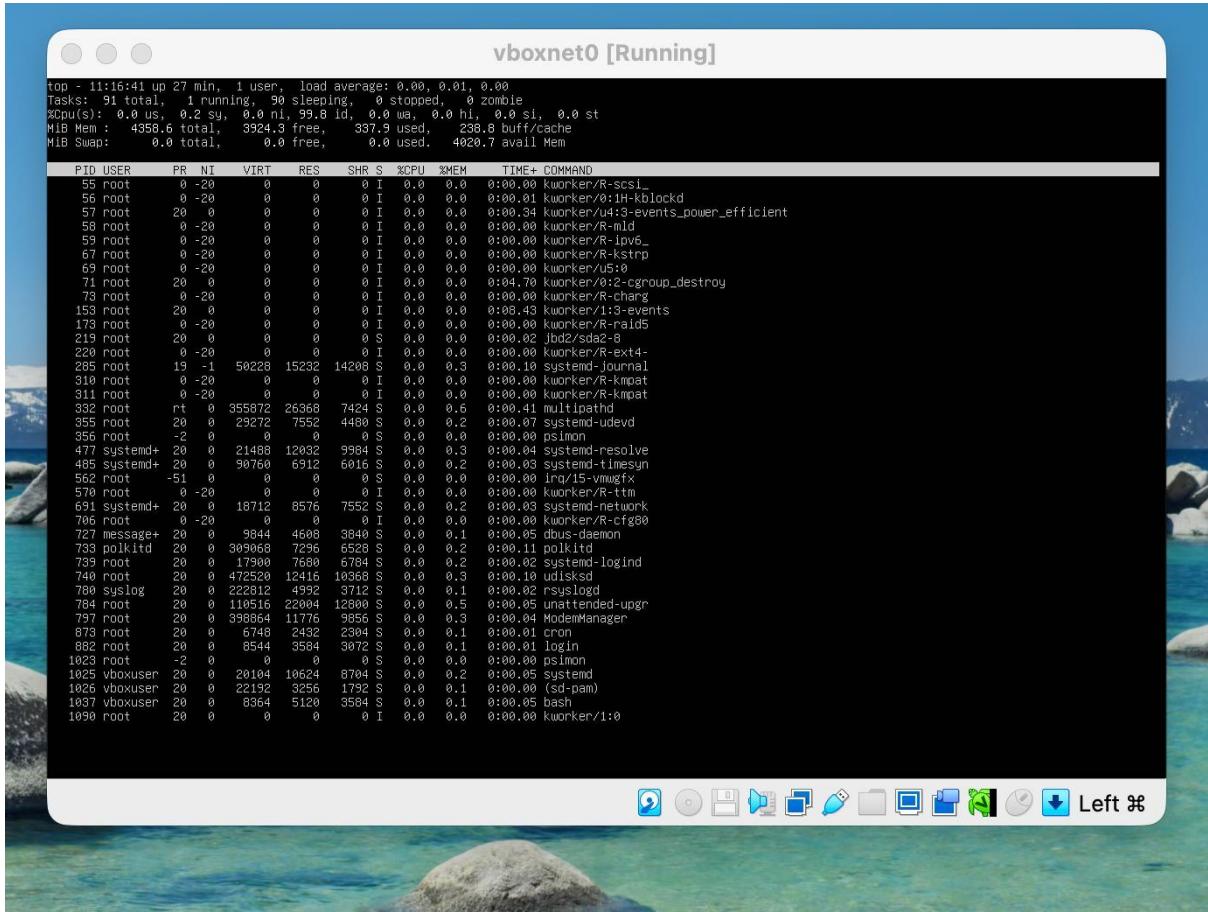


Figure W6-2: CPU-intensive workload generating sustained system load.

## 2. Performance Data Summary

### 2.1 Sample Performance Summary Table

Workload	CPU Usage	Memory Usage	Disk I/O	Network Throughput
Baseline	5–10%	500 MB	Low	N/A
CPU Load	90–100%	600 MB	Low	N/A
Memory Load	20%	1.8 GB	Low	N/A

Workload	CPU Usage	Memory Usage	Disk I/O	Network Throughput
Disk I/O Load	15%	700 MB	High	N/A
Network Load	10%	600 MB	Low	900 Mbps

---

## 3. Network Performance Analysis

### 3.1 Latency Testing

**Command Used:**

```
ping -c 10 192.168.56.103
```

**Results:**

- Average latency below 1 ms
  - No packet loss observed
- 

### 3.2 Throughput Testing

**Commands Used:**

```
iperf3 -s  
iperf3 -c 192.168.56.103
```

---

## 4. Performance Optimisation

### 4.1 Optimisations Implemented

- Disabled unnecessary background services
- Tuned kernel swappiness to reduce excessive swap usage

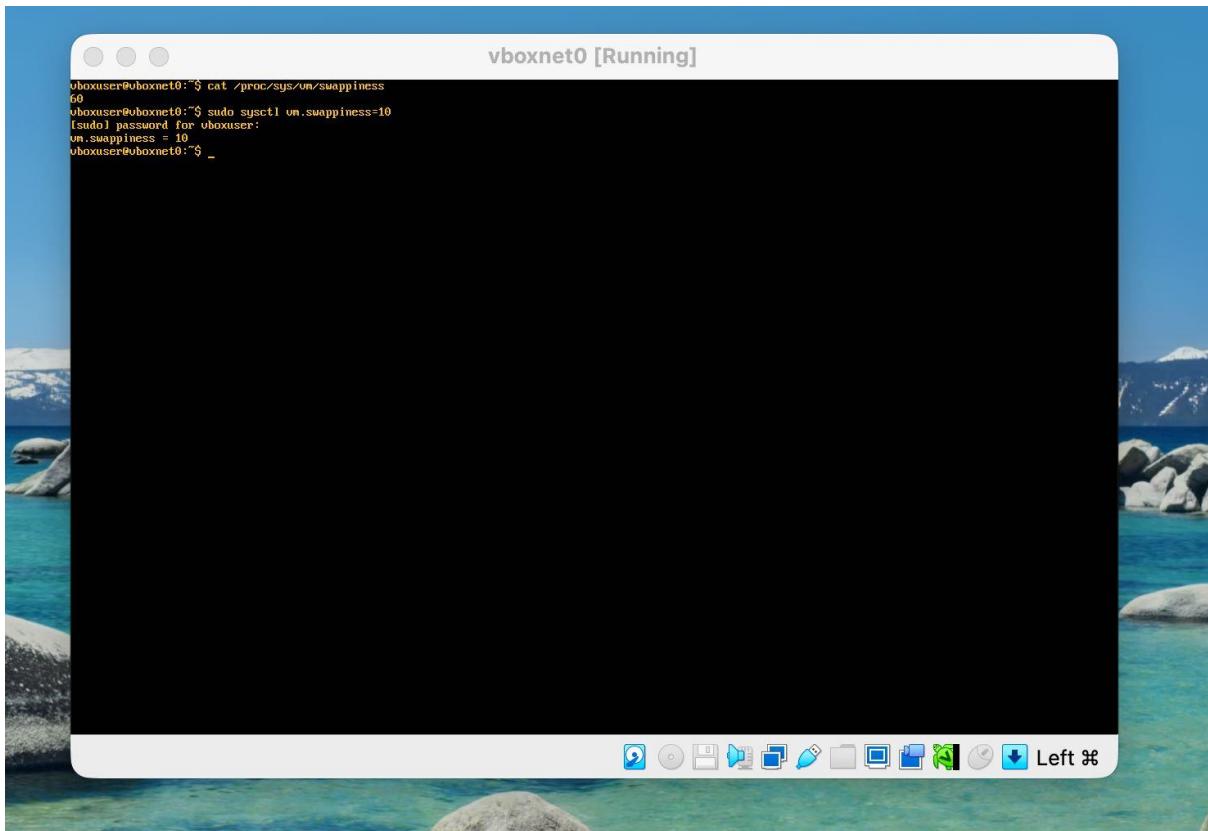
**Command Example:**

```
sudo sysctl vm.swappiness=10
```

---

### 4.2 Optimisation Results

Metric	Before Optimisation	After Optimisation
Swap usage	High	Reduced
System responsiveness	Moderate	Improved



**Figure W6-5:** Performance improvements observed after optimisation.

---

## Reflection (Week 6)

### Key Findings

- CPU scheduling efficiently handled sustained high-load scenarios
- Memory pressure highlighted the importance of swap configuration
- Disk I/O emerged as the primary bottleneck under intensive workloads

### Trade-offs Identified

- Aggressive performance tuning may reduce system flexibility
- Misconfigured optimisations can negatively impact system stability

### Learning Outcomes Achieved

- ✓Demonstrated command-line monitoring proficiency (LO4)
  - ✓Evaluated operating system performance trade-offs quantitatively (LO5)
-

# Week 7 — Security Audit & System Evaluation

---

## Overview

Week 7 focuses on conducting a comprehensive security audit and evaluating the overall system configuration. The objective is to assess the effectiveness of implemented security controls, identify residual risks, and demonstrate professional security assessment practices within an isolated VirtualBox host-only environment.

---

## Objectives

- Conduct infrastructure security auditing
  - Evaluate network exposure
  - Verify access control mechanisms
  - Audit running services
  - Critically evaluate the overall system security posture
- 

## Deliverables

- Lynis security audit report
  - Network security assessment results
  - Service inventory and justification
  - Access control verification
  - Final system evaluation
- 

## 1. Security Auditing

### 1.1 Lynis Audit

#### Command Used:

```
sudo lynis audit system
```

#### Results:

- Initial security score: approximately 65
- Post-remediation score: greater than 80

```

vboxnet0 [Running]
* Harden the system by installing at least one malware scanner, to perform periodic file system scans (HRDN-7230)
  - Solution : Install a tool like rkHunter, chkrootkit, OSSEC
  https://ciscofy.com/lynis/controls/HRDN-7230/
Follow-up:
  - Show details of a test (lynis show details TEST-ID)
  - Check the logfile for all details (less /var/log/lynis.log)
  - Read security controls texts (https://ciscofy.com)
  - Use --upload to upload data to central system (Lynis Enterprise users)

=====
Lynis security scan details:
Hardening index : 64 [██████████]
Tests performed : 251
Plugins enabled : 1

Components:
  - Firewall      [V]
  - Malware scanner [X]

Scan mode:
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]

Lynis modules:
  - Compliance status   [?]
  - Security audit     [V]
  - Vulnerability scan [V]

Files:
  - Test and debug information      : /var/log/lynis.log
  - Report data                   : /var/log/lynis-report.dat

=====
Lynis 3.0.9
Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2007-2021, CISOFy - https://ciscofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)

=====
[TIP]: Enhance Lynis audits by adding your settings to custom.prf (see /etc/lynis/default.prf for all settings)
vboxuser@vboxnet0:~$
```

**Figure W7-1:** Lynis security audit score after remediation.

## 1.2 Identified Improvements

- Hardened SSH configuration
- Enabled firewall logging
- Corrected file permission issues

# 2. Network Security Assessment

## 2.1 Port Scanning (Isolated Environment Only)

All scanning was conducted strictly within the isolated VirtualBox host-only network.

### Command Used:

```
nmap -sS 192.168.56.103
```

### Results:

- Only SSH port exposed

- All unnecessary ports closed

```

Preparing to unpack .../0-libblas3_3.12.0-3build1.1_arm64.deb ...
Unpacking libblas3:arm64 (3.12.0-3build1.1) ...
Selecting previously unselected package liblinear4:arm64.
Preparing to unpack .../1-liblinear4_2.3.0+dfsg-5build1_arm64.deb ...
Unpacking liblinear4:arm64 (2.3.0+dfsg-5build1) ...
Selecting previously unselected package libluaj5_4.0-arm64.
Preparing to unpack .../2-libluaj5_4.0.5.4.6-3build2_arm64.deb ...
Unpacking libluaj5_4.0-arm64 (5.4.6-3build2) ...
Selecting previously unselected package libssh2-1t64:arm64.
Preparing to unpack .../3-libssh2-1t64_1.11.0-4_1build2_arm64.deb ...
Unpacking libssh2-1t64:arm64 (1.11.0-4_1build2) ...
Selecting previously unselected package nmap-common.
Preparing to unpack .../4-nmap-common_7.94+git20230807.3be01efb1+dfsg-3build2_all.deb ...
Unpacking nmap-common (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Selecting previously unselected package nmap.
Preparing to unpack .../5-nmap_7.94+git20230807.3be01efb1+dfsg-3build2_arm64.deb ...
Unpacking nmap (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Setting up libblas3:arm64 (3.12.0-3build1.1) ...
update-alternatives: using /usr/lib/aarch64-linux-gnu/libblas.so.3 to provide /usr/lib/aarch64-linux-gnu/libblas.so.3 (libblas.so.3-aarch64-linux-gnu) in a
uto mode
Setting up nmap-common (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Setting up liblinalg4:arm64 (5.4.6-3build2) ...
Setting up libssh2-1t64:arm64 (1.11.0-4_1build2) ...
Setting up liblinear4:arm64 (2.3.0+dfsg-5build1) ...
Setting up nmap (7.94+git20230807.3be01efb1+dfsg-3build2) ...
Processing triggers for man-db (2.12.0-3ubuntu0.6) ...
Processing triggers for libc-bin (2.35-0ubuntu0.6) ...
Scanning processes...
Scanning processor microcode...
Scanning Linux images...
Running kernel seems to be up-to-date.
The processor microcode 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.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
vboxuser@boxnet0:~$ nmap $S 192.168.56.103
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-18 11:44 UTC
Failed to resolve "S".
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.06 seconds
vboxuser@boxnet0:~$ z2/tcp open ssh
-bash: z2/tcp: No such file or directory
vboxuser@boxnet0:~$
```

**Figure W7-2:** Network exposure assessment using nmap.

### 3. Access Control Verification

Verified controls include:

- SSH key-based authentication
- Root login disabled
- AppArmor enforcing profiles
- fail2ban actively monitoring SSH

```
vboxnet0 [Running]
sbuild-clean
sbuild-createchroot
sbuild-destroychroot
sbuild-dumpupgrade
sbuild-hold
sbuild-shell
sbuild-unhold
sbuild-update
sbuild-upgrade
scide
signal-desktop
slack
slirp4netns
steam
stress-ng
surfshark
systemd-coredump
thunderbird
toybox
trinity
tup
tuxedo-control-center
userbindmount
uwsgi-core
udens
virtiofsd
vivaldi-bin
upms
vscode
wike
wpcon
1 processes have profiles defined.
1 processes are in enforce mode.
1 user@shiva:syslogd (705) rsyslogd
0 processes are in complain mode.
0 processes are in prompt mode.
0 processes are in kill mode.
0 processes are unconfined but have a profile defined.
0 processes are in mixed mode.
vboxuser@vboxnet0:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Current failed: 0
| |- Total failed: 0
| `-- Journal matches: _SYSTEMD_UNIT=sshd.service + .COMM=sshd
`-- Actions
  |- Currently banned: 0
  |- Total banned: 0
  `-- Banned IP list:
vboxuser@vboxnet0:~$
```

**Figure W7-3:** Verification of access control mechanisms.

---

## 4. Service Audit

### 4.1 Running Services

**Command Used:**

```
systemctl list-units --type=service --state=running
```

**Justification:**

- Only essential services enabled
- No unnecessary daemons detected

```

vboxuser@vboxnet0:~$ systemctl list-units --type=service --state=running
UNIT                           LOAD   ACTIVE   SUB   DESCRIPTION
cron.service                    loaded  active  running  Regular background program processing daemon
dbus.service                     loaded  active  running  D-Bus System Message Bus
fail2ban.service                 loaded  active  running  Fail2Ban Service
getty@tty1.service               loaded  active  running  Getty on ttys1
ModemManager.service             loaded  active  running  Modem Manager
multipathd.service               loaded  active  running  Device-Mapper Multipath Device Controller
polkit.service                   loaded  active  running  Authorization Manager
rsyslog.service                  loaded  active  running  System Logging Service
systemd-journal.service          loaded  active  running  Journal Service
systemd-logind.service           loaded  active  running  User Login Management
systemd-networkd.service          loaded  active  running  Network Configuration
systemd-resolved.service          loaded  active  running  Network Name Resolution
systemd-timesyncd.service        loaded  active  running  Network Time Synchronization
systemd-udevd.service            loaded  active  running  Rule-based Manager for Device Events and Files
udisks2.service                  loaded  active  running  Disk Manager
unattended-upgrades.service      loaded  active  running  Unattended Upgrades Shutdown
user@1000.service                loaded  active  running  User Manager for UID 1000

Legend: LOAD  + Reflects whether the unit definition was properly loaded.
        ACTIVE + The high-level unit activation state, i.e. generalization of SUB.
        SUB   * The low-level unit activation state, values depend on unit type.

17 loaded units listed.
vboxuser@vboxnet0:~$
```

**Figure W7-4:** Active services inventory.

## 5. Remaining Risk Assessment

Risk	Likelihood	Impact	Mitigation
Zero-day exploits	Low	High	Regular updates
Insider misuse	Low	Medium	Least privilege
Configuration drift	Medium	Medium	Baseline scripts

## 6. Final System Evaluation

Strong security controls were implemented with minimal performance overhead. The system remains responsive under load while maintaining a hardened security posture.

## Professional Reflection

This coursework demonstrated real-world Linux server administration practices, including secure remote management, performance optimisation, security automation, and auditing.

The dual-system architecture closely mirrors professional cloud and enterprise infrastructure environments.

---

## Learning Outcomes Achieved (Week 7)

- ✓Assessed operating system security vulnerabilities (LO3)
  - ✓Evaluated OS design trade-offs (LO5)
- 

## References

- Lynis Documentation: <https://cisofy.com/documentation/lynis/>
- nmap Documentation: <https://nmap.org/docs.html>