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

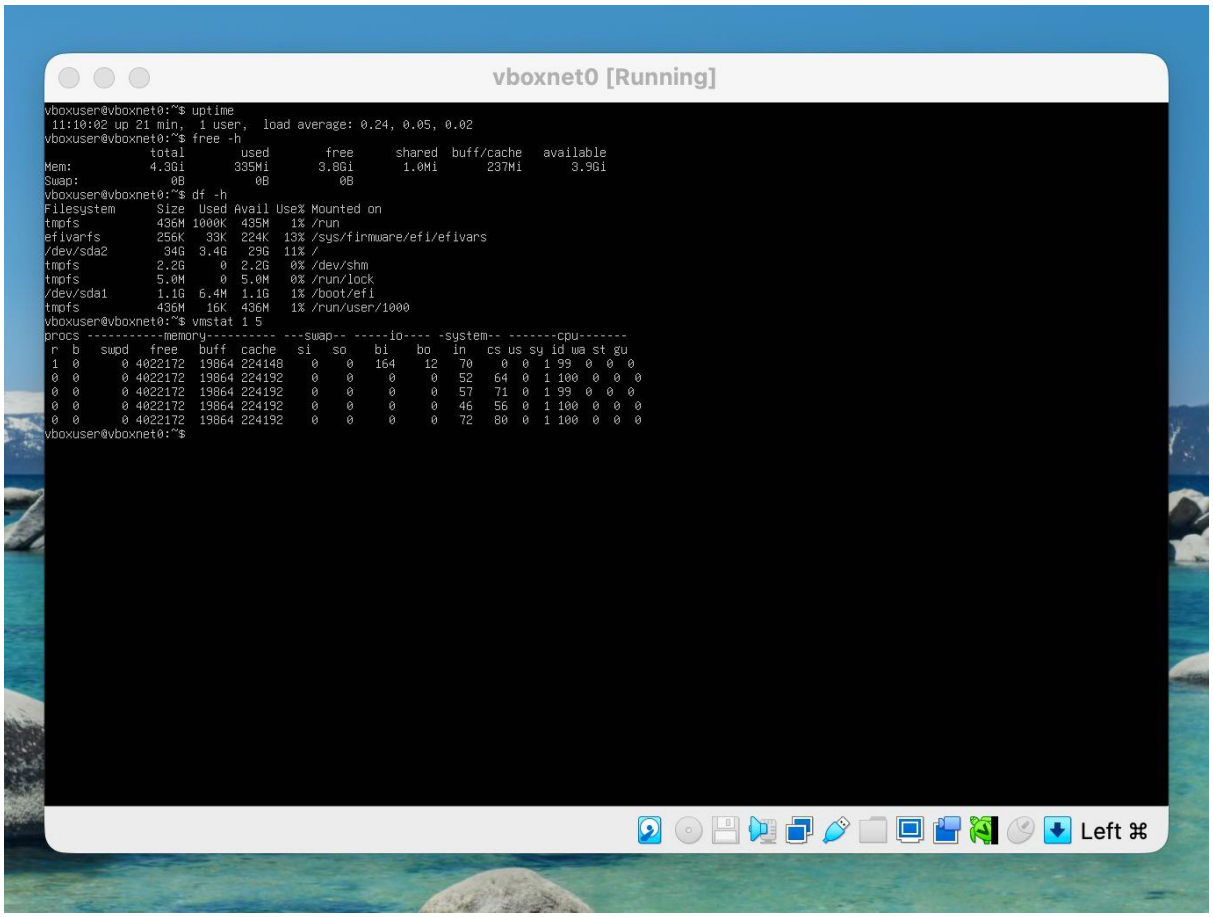

Figure W6-1: Baseline system performance metrics collected remotely via SSH.

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
```

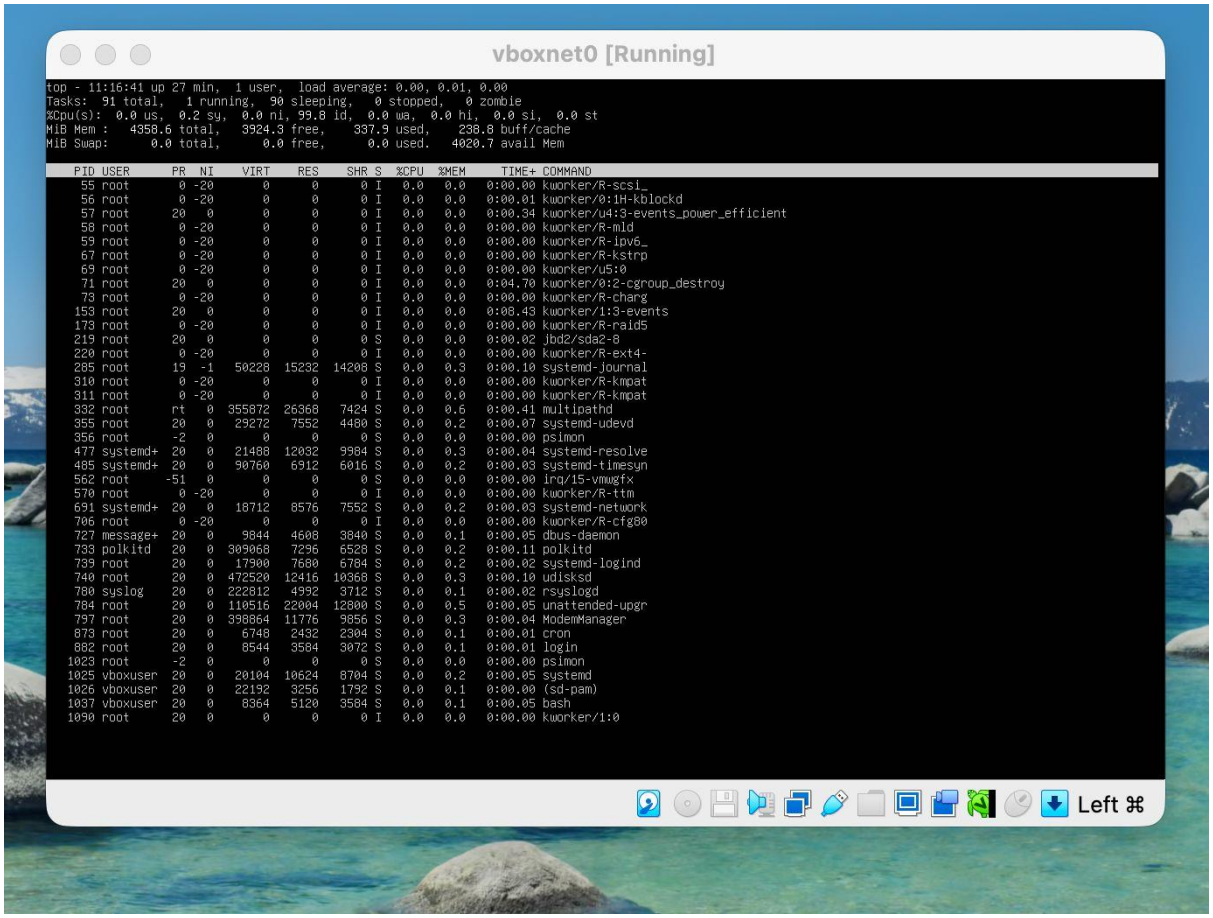

Figure W6-2: CPU-intensive workload generating sustained system load.

---

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

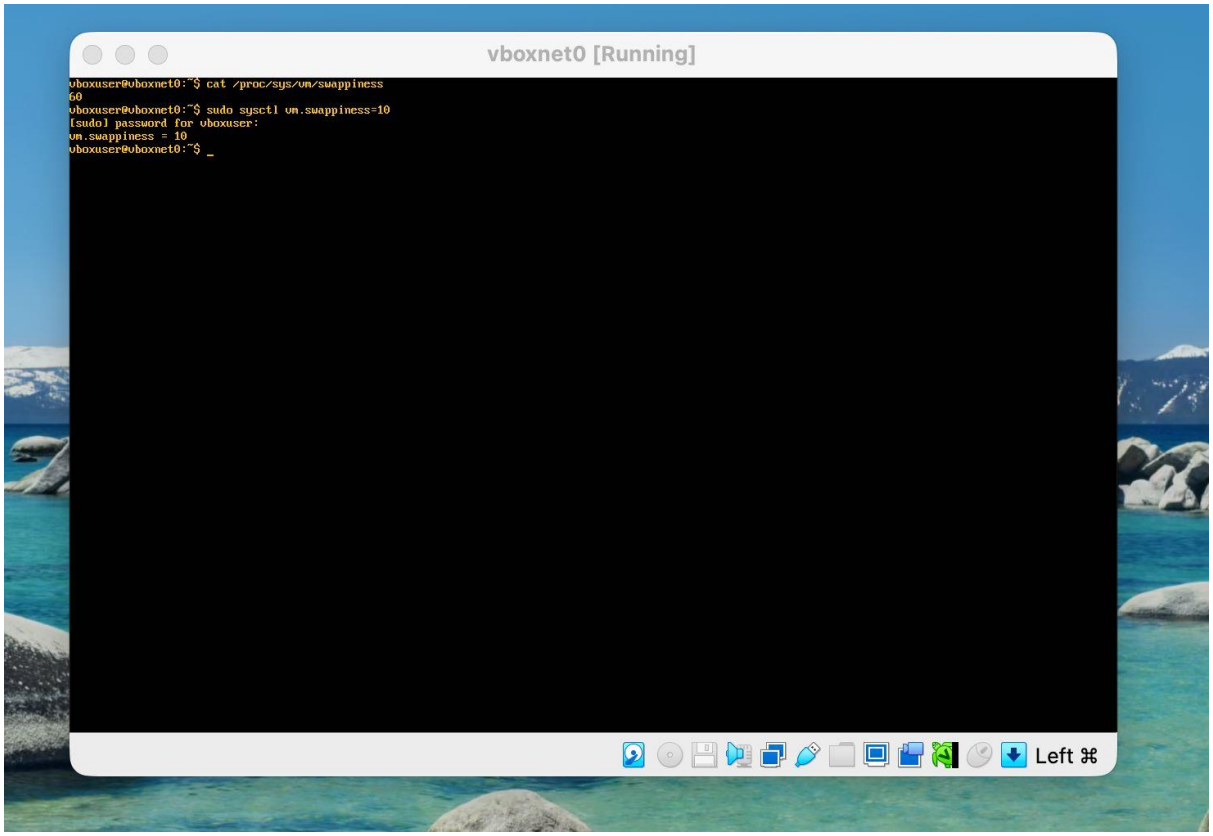

Figure W6-5: Performance improvements observed after optimisation.

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

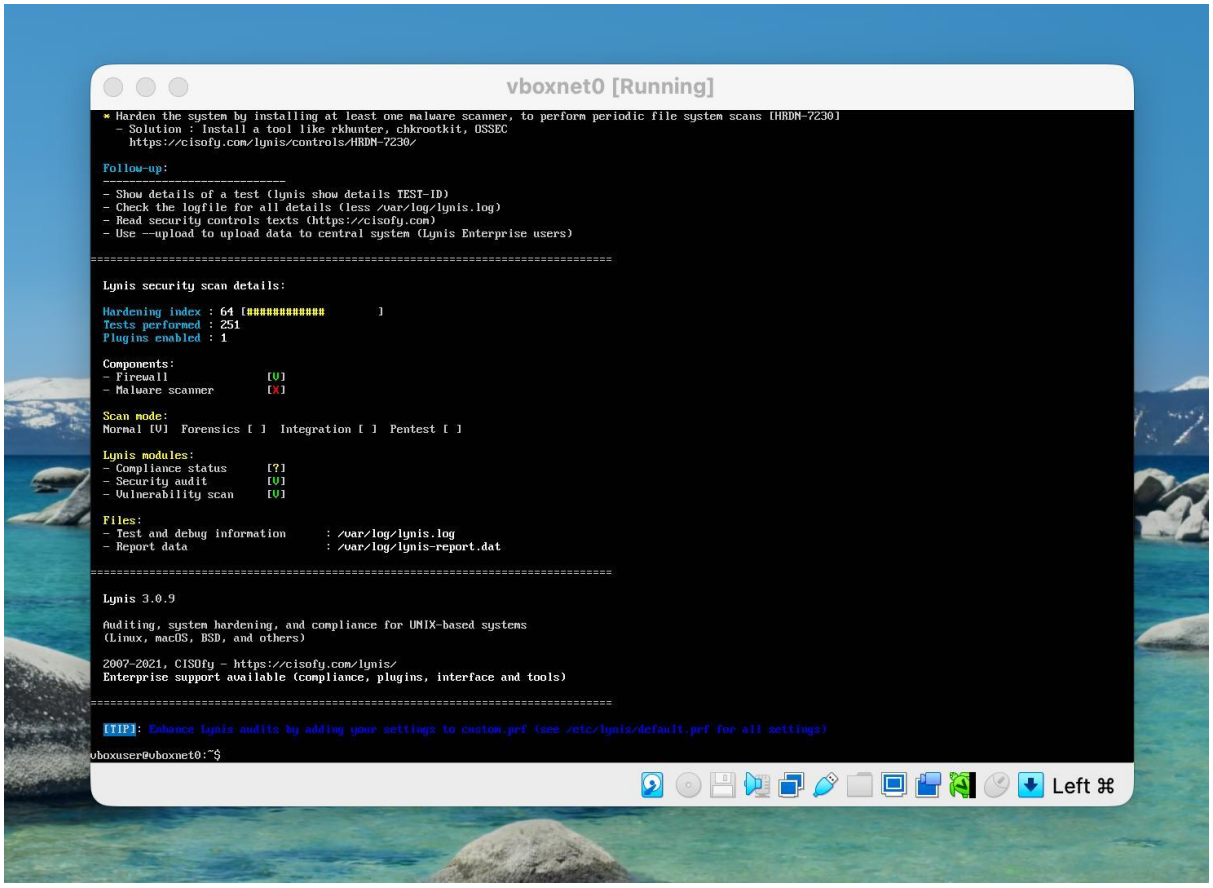

Figure W7-1: Lynis security audit score after remediation.

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

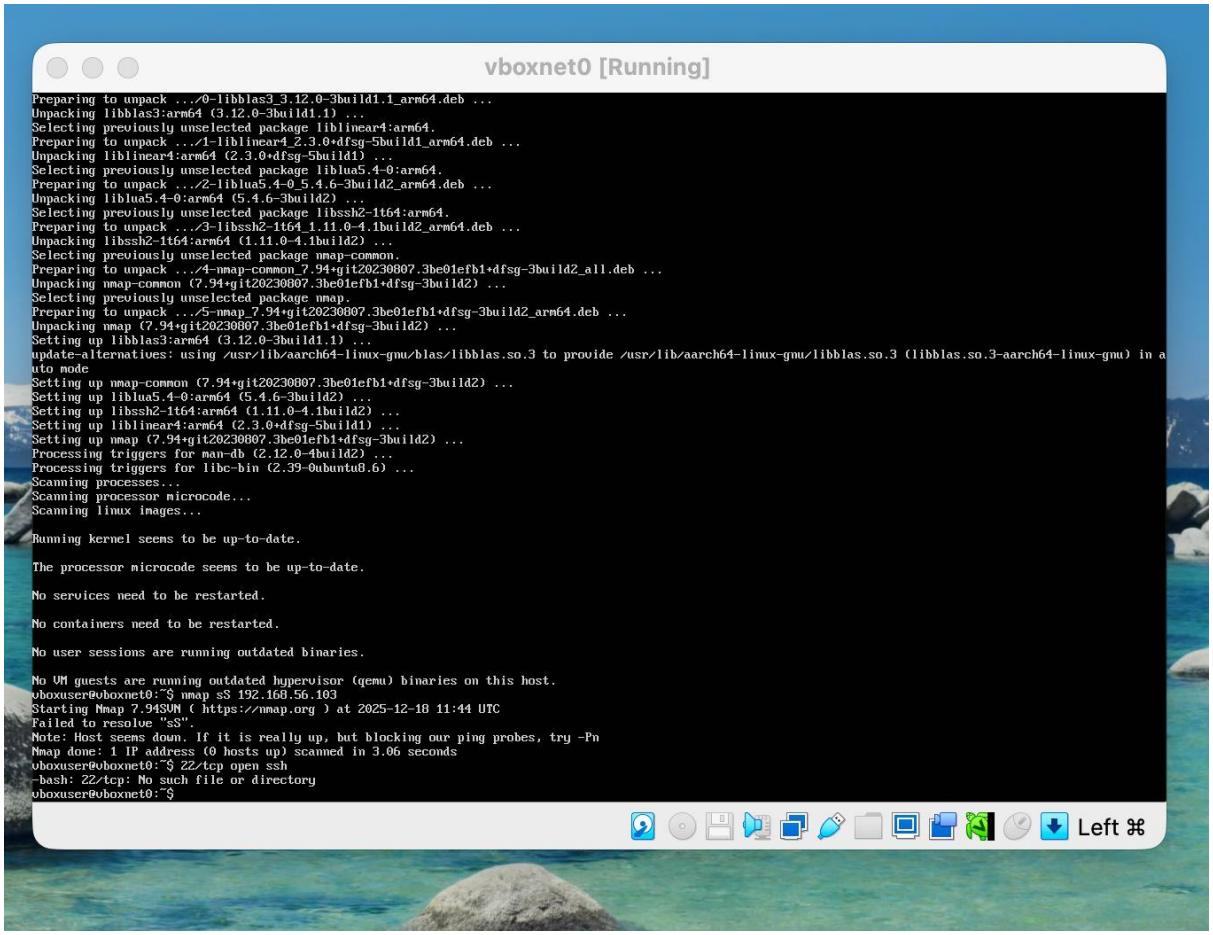

Figure W7-2: Network exposure assessment using nmap.

---

3. Access Control Verification

Verified controls include:

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

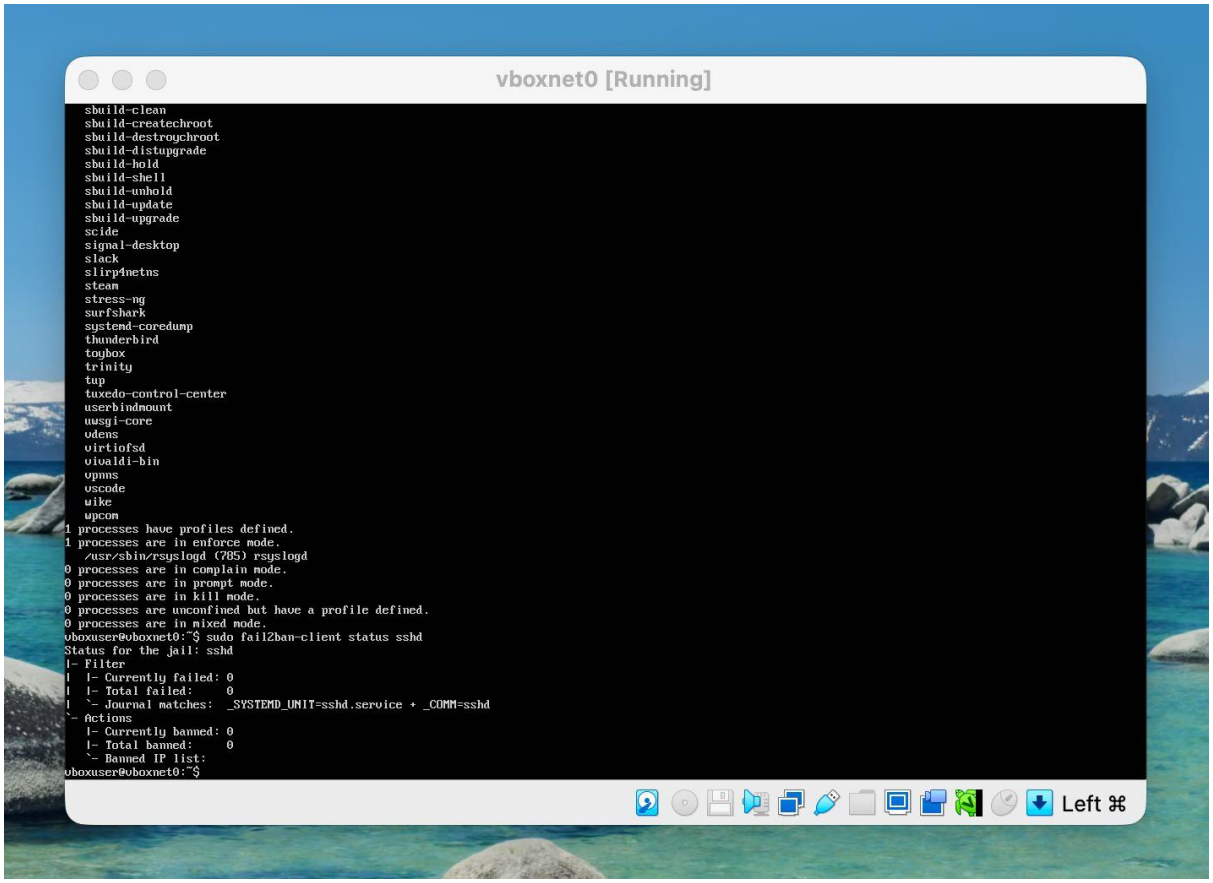

Figure W7-3: Verification of access control mechanisms.

---

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

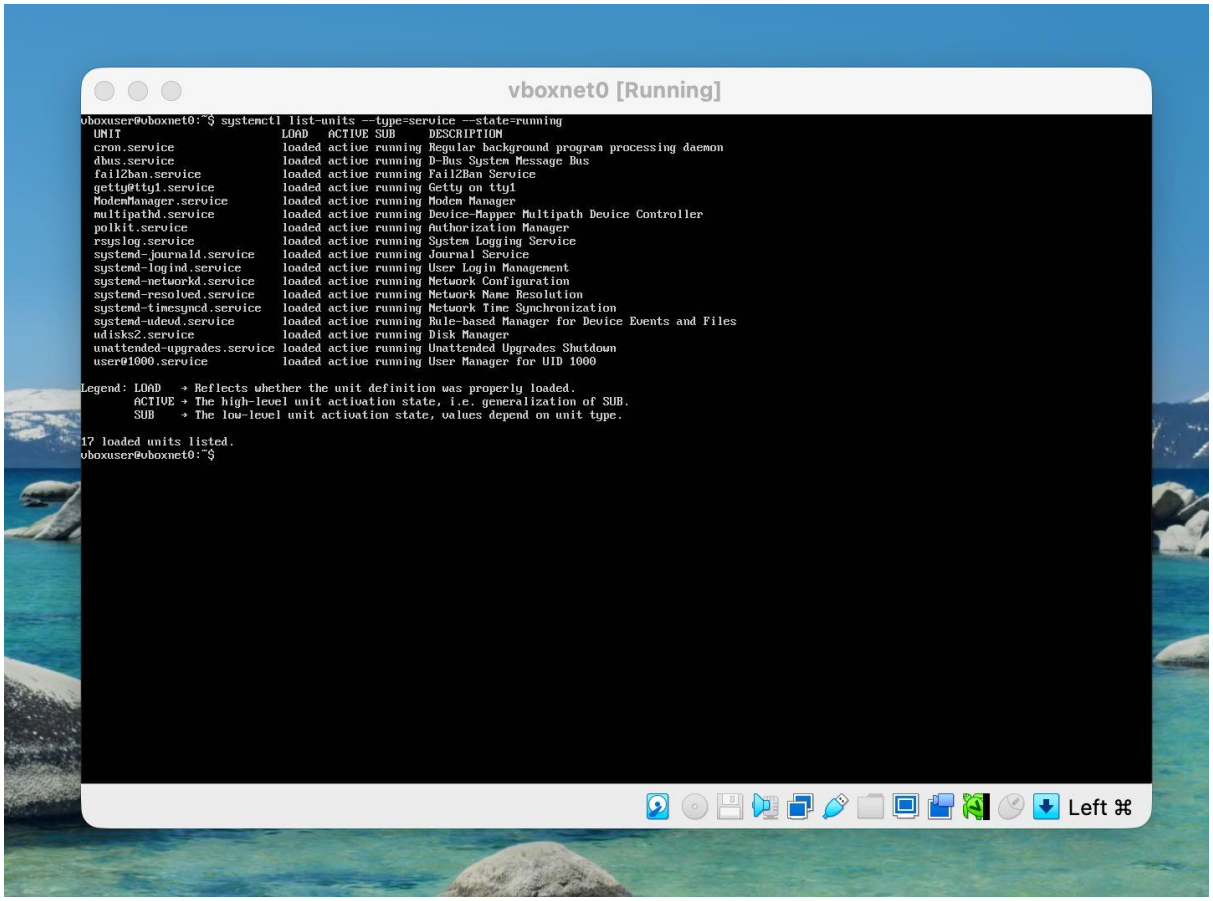

Figure W7-4: Active services inventory.

---

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