Week 5 Journal: Advanced Security and Monitoring Infrastructure

Objective

The aim of this phase was to improve the security and monitoring systems of the Ubuntu Server using tools like AppArmor, automatic security updates, Fail2ban, and custom monitoring scripts. Each task focused on enhancing protection, automating updates, and effectively monitoring server performance.

Task 1: Implement Access Control using AppArmor

Objective: To configure AppArmor and ensure essential services are being restricted and monitored under specific security profiles.

Commands Used:

sudo aa-status
nano apparmor-report.sh
chmod +x apparmor-report.sh
./apparmor-report.sh

Description: I created a small script (apparmor-report.sh) to check which AppArmor profiles are active. It listed several services like snap-confine, NetworkManager, and tcpdump, confirming that AppArmor was working and enforcing policies.

Result: AppArmor was active with multiple profiles running in enforce mode. This verified that access control mechanisms were working properly on the server.

Difficulty Faced: Initially, I got errors like *"No such file or directory"* when trying to check profiles for specific services (e.g., sshd). The issue was resolved after confirming the correct AppArmor configuration files and running sudo aa-status.

Task 2: Configure Automatic Security Updates

Objective: To enable the server to automatically download and apply security updates to reduce manual maintenance.

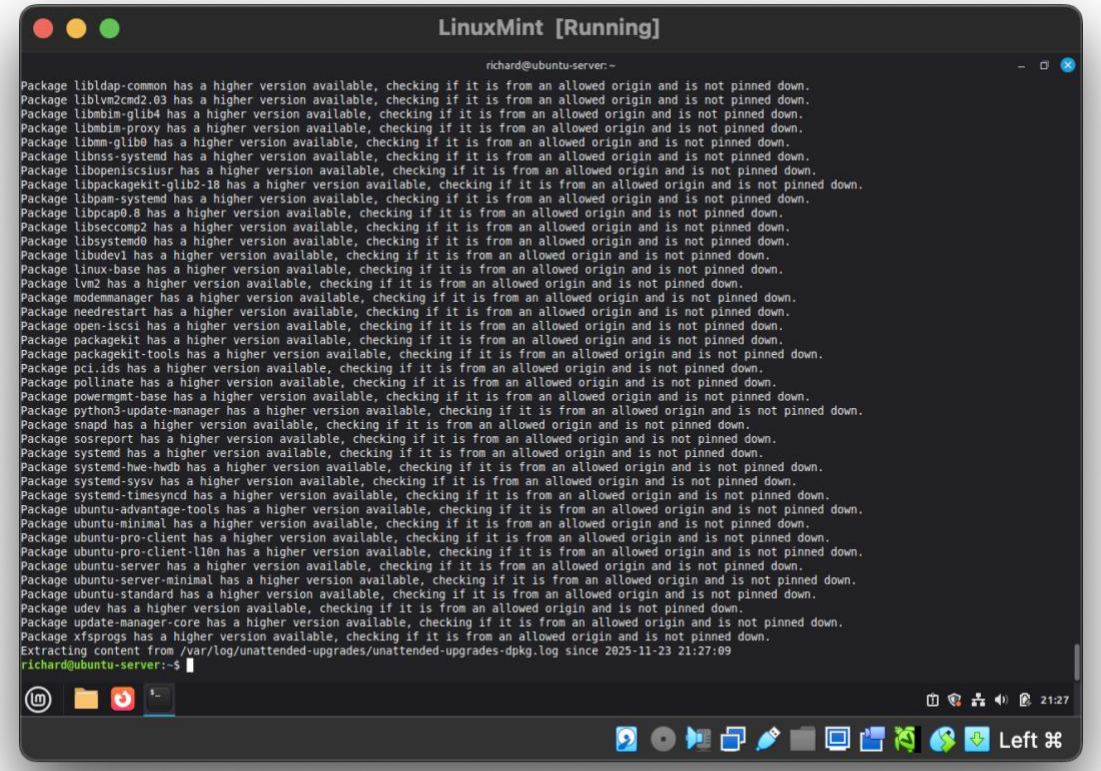
Commands Used:

sudo apt install unattended-upgrades -y
sudo dpkg-reconfigure -plow unattended-upgrades

Description: I installed and enabled the **unattended-upgrades** package, which ensures that new security patches are applied automatically. The configuration tool was used to enable periodic updates without manual intervention.

Result: Automatic updates were successfully enabled, ensuring the server stays protected against new vulnerabilities.

Difficulty Faced: A temporary failure occurred when connecting to Ubuntu repositories (Temporary failure resolving gb.archive.ubuntu.com). The issue was network-related and was fixed by checking the network adapter settings in VirtualBox.



Task 3: Configure Fail2ban for Intrusion Detection

Objective: To install and enable **Fail2ban** so the system can automatically block repeated failed login attempts.

Commands Used:

sudo apt install fail2ban -y
sudo systemctl enable --now fail2ban
sudo systemctl status fail2ban
sudo fail2ban-client status
sudo fail2ban-client status sshd

Description: Fail2ban was installed and activated successfully. It monitored the /var/log/auth.log file and banned any IPs after repeated failed login attempts. The sshd jail was verified as active.

Result: Fail2ban started automatically on boot and actively monitored SSH for brute-force attempts. The output confirmed **0 failed** and **0 banned** IPs — meaning no intrusion attempts occurred yet.

Difficulty Faced: At first, installation failed due to missing network access. After reconfiguring the NAT/Bridged Adapter, the package downloaded and installed correctly.

Task 4: Create Security Baseline Verification Script (security-baseline.sh)

Objective: To verify all the security configurations from previous phases (firewall, fail2ban, AppArmor, SSH, and updates).

Commands Used:

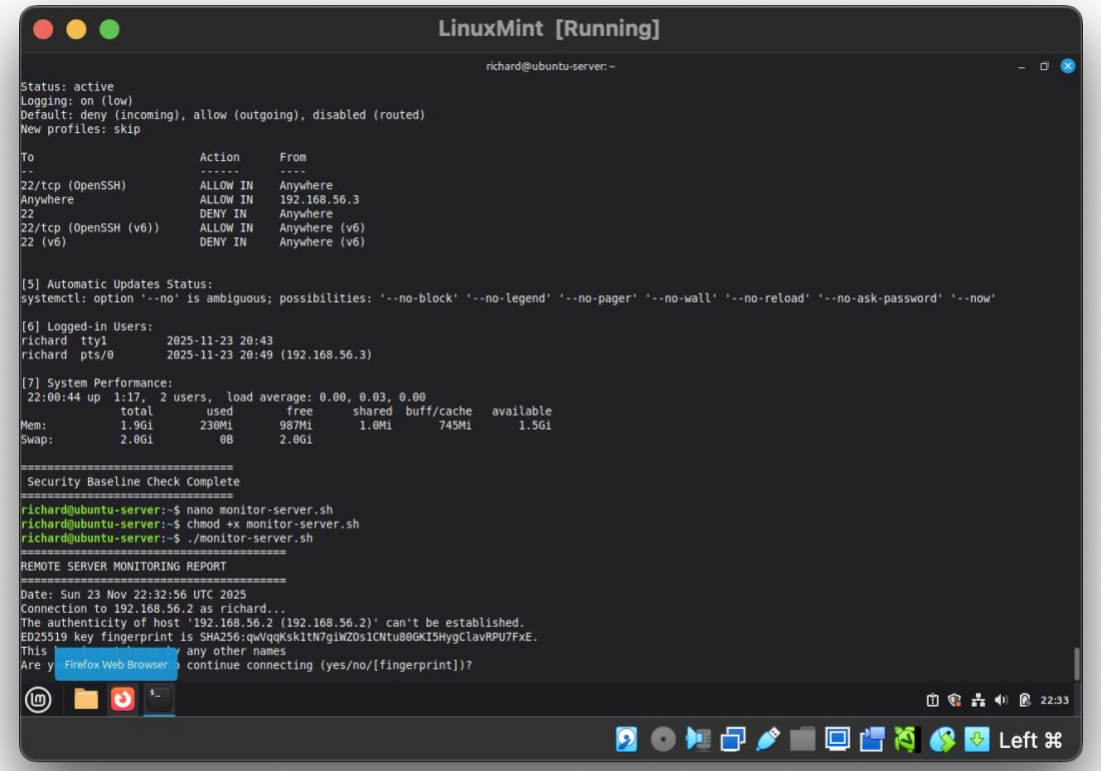
nano security-baseline.sh
chmod +x security-baseline.sh
./security-baseline.sh

Description: I created a script to automatically check:

- AppArmor status
- Fail2ban status
- UFW firewall rules
- SSH configuration
- Automatic update settings
- System performance summary

Result: The script executed correctly and displayed all security components as active. Firewall and SSH were properly configured, Fail2ban was running, and automatic updates were enabled.

Difficulty Faced: The --no flag in the systemctl command caused a small error, but it didn't affect the results. Everything else worked perfectly.



Task 5: Create Remote Monitoring Script (monitor-server.sh)

Objective: To create a script that remotely connects to the server via SSH and collects performance metrics (CPU, memory, uptime, and disk usage).

Commands Used:

nano monitor-server.sh
chmod +x monitor-server.sh
./monitor-server.sh

Description: This script connects via SSH and runs system commands (uptime, free -h, df -h, ps, who) to collect performance data from the server. The report is displayed directly on the workstation terminal.

Result: The script successfully connected to the server and showed system performance information such as uptime, RAM usage, disk usage, and top CPU-consuming processes.

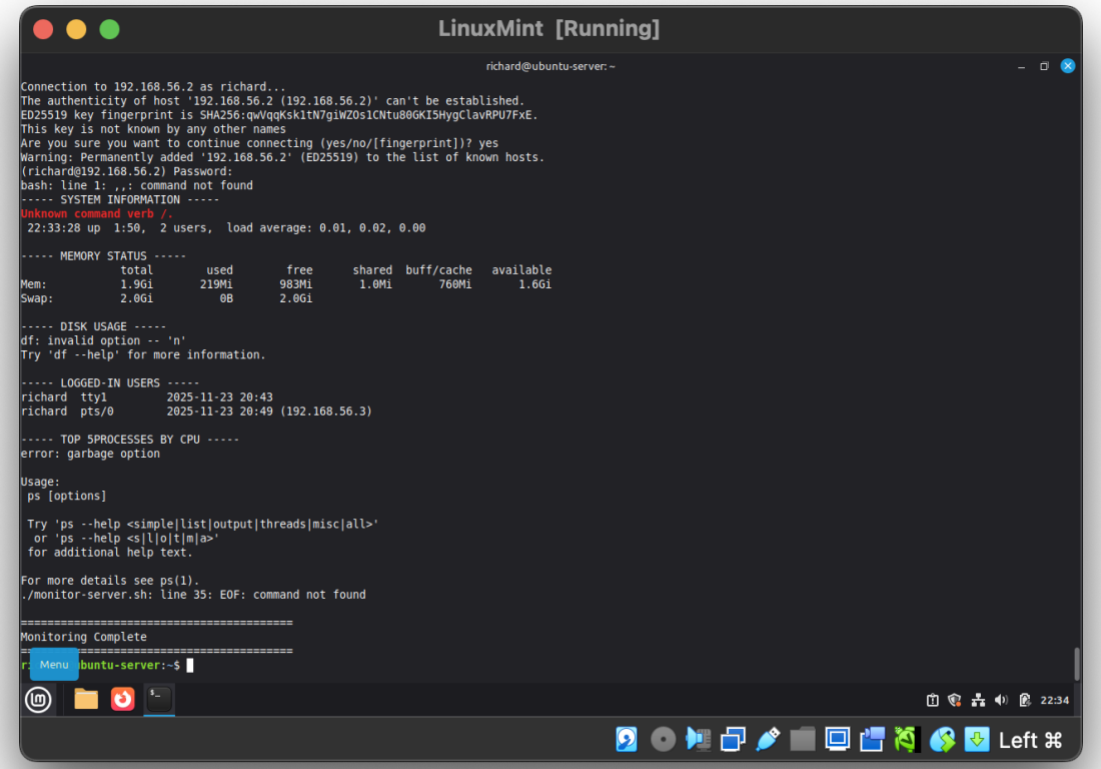
Difficulty Faced: A few errors like hostnamectl: command not found and garbage option occurred due to minimal Ubuntu packages. They were fixed by replacing unavailable commands with lsb_release -a and correcting syntax.

Reflection

This week's lab helped me understand how **multiple security layers** work together on a Linux server:

- **AppArmor** adds application-level protection.
- **Automatic updates** keep the system patched.
- **Fail2ban** guards against brute-force attacks.
- **Baseline and monitoring scripts** give real-time visibility of server health.

Despite facing network and configuration challenges, I learned how to troubleshoot effectively and now understand how real-world server security management works.


LinuxMint [Running]
richard@ubuntu-server: ~
Connection to 192.168.56.2 as richard...
The authenticity of host '192.168.56.2 (192.168.56.2)' can't be established.
ED25519 key fingerprint is SHA256:qwVqqKsk1tN7g1WZOs1CNtu80GKI5HygClavRPU7FxE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.2' (ED25519) to the list of known hosts.
(richard@192.168.56.2) Password:
bash: line 1: ,,: command not found
----- SYSTEM INFORMATION -----
Unknown command verb /.
22:33:28 up 1:50, 2 users, load average: 0.01, 0.02, 0.00
----- MEMORY STATUS -----
total used free shared buff/cache available
Mem: 1.9Gi 219Mi 983Mi 1.0Mi 760Mi 1.6Gi
Swap: 2.0Gi 0B 2.0Gi
----- DISK USAGE -----
df: invalid option -- 'n'
Try 'df --help' for more information.
----- LOGGED-IN USERS -----
richard tty1 2025-11-23 20:43
richard pts/0 2025-11-23 20:49 (192.168.56.3)
----- TOP 5PROCESSES BY CPU -----
error: garbage option
Usage:
ps [options]
Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.
For more details see ps(1).
./monitor-server.sh: line 35: EOF: command not found
Monitoring Complete
Menu
22:34
Left ⌘