Week 5: Advanced Security and Monitoring Infrastructure

1. Aim of the Week

The aim of Week 5 was to implement advanced security controls on the Ubuntu server and develop basic automation and monitoring capabilities. This week built on the foundational security implemented in Week 4 by adding access control enforcement, intrusion detection, automatic security updates, and scripted verification and monitoring. All configurations and scripts were executed remotely via SSH in line with professional system administration practices.

2. Mandatory Access Control with AppArmor

Ubuntu uses AppArmor as its default Mandatory Access Control (MAC) system. AppArmor restricts what applications are allowed to access, even if the application or service is compromised.

AppArmor utilities were installed, and the service status was verified to confirm that the AppArmor kernel module was loaded and active. The AppArmor status command was used to display all loaded profiles and identify which profiles were running in enforce mode.

The output confirmed that multiple profiles were actively enforcing restrictions, demonstrating that AppArmor was functioning correctly on the system.

3. Automatic Security Updates

Automatic security updates were configured using the unattended-upgrades service. This ensures that critical security patches are applied automatically without requiring manual administrator intervention.

The unattended-upgrades service was verified as active and running. This provides ongoing protection against newly discovered vulnerabilities by ensuring the system remains up to date with security fixes.

Automatic updates reduce the window of exposure between vulnerability disclosure and patch installation, which is essential for maintaining system security.

4. Intrusion Detection with fail2ban

fail2ban was installed to provide intrusion detection and automated response to suspicious activity. fail2ban monitors log files for repeated failed authentication attempts and temporarily blocks offending IP addresses.

The fail2ban service was enabled and verified as running. This adds an additional layer of protection to the SSH service by mitigating brute-force login attempts.

By automatically responding to suspicious behaviour, fail2ban reduces the likelihood of successful unauthorized access.

5. Security Baseline Verification Script (security-baseline.sh)

A security baseline verification script named security-baseline.sh was created and executed on the server via SSH. The purpose of this script is to automatically verify that the security controls from Weeks 4 and 5 are correctly implemented.

The script checks:

- SSH service availability and hardening settings
- Firewall status and active rules
- AppArmor service status and enforced profiles
- Automatic security update configuration
- fail2ban service status and SSH monitoring

Each check outputs a clear status message indicating whether the configuration is correctly applied. All commands in the script are commented line by line to explain their purpose and behaviour.

This script allows quick validation of the server's security posture and ensures consistency across configuration changes.

6. Remote Monitoring Script (monitor-server.sh)

A remote monitoring script named monitor-server.sh was created on the workstation. This script connects to the server via SSH and collects key performance metrics without direct access to the server console.

The script retrieves:

- Hostname and uptime
- CPU usage snapshot
- Memory usage
- Disk usage
- Network interface statistics

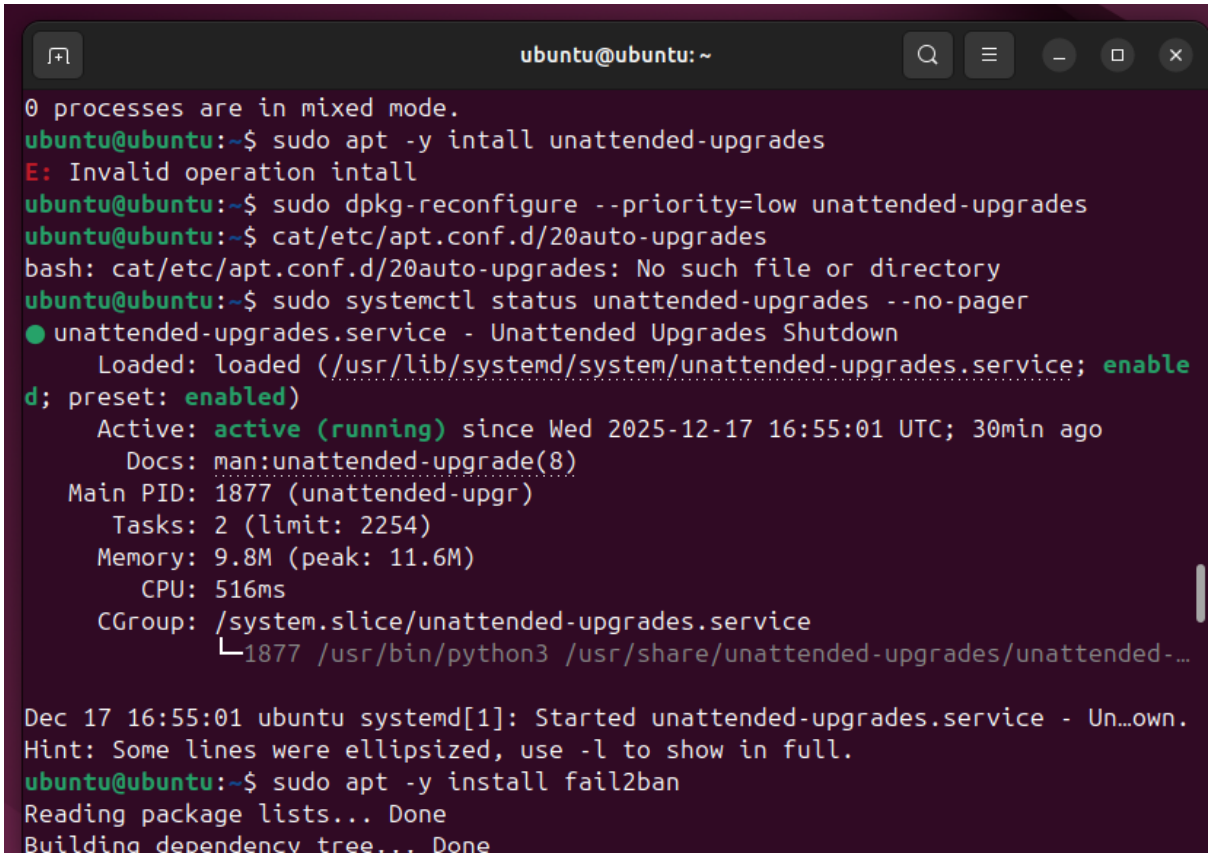
All commands are executed remotely using SSH, demonstrating real-world monitoring techniques. The script output provides a concise overview of system health and resource utilisation.

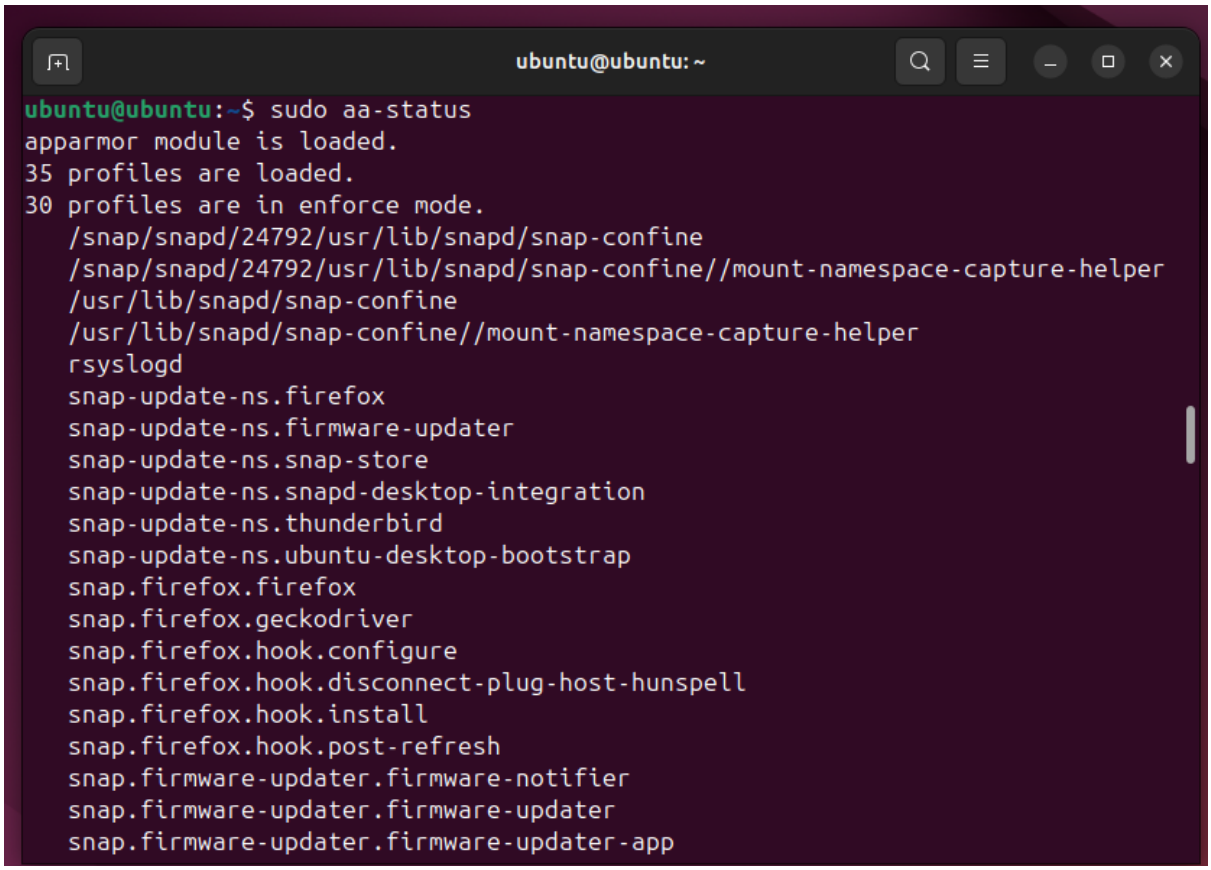
Each command in the script is documented with comments to clearly explain its function.

7. Reflection

Week 5 demonstrated how security and monitoring can be automated to improve reliability and efficiency. Implementing AppArmor showed how Mandatory Access Control limits application behaviour, while fail2ban provided automated protection against brute-force attacks. Automatic security updates reduced the need for manual patch management.

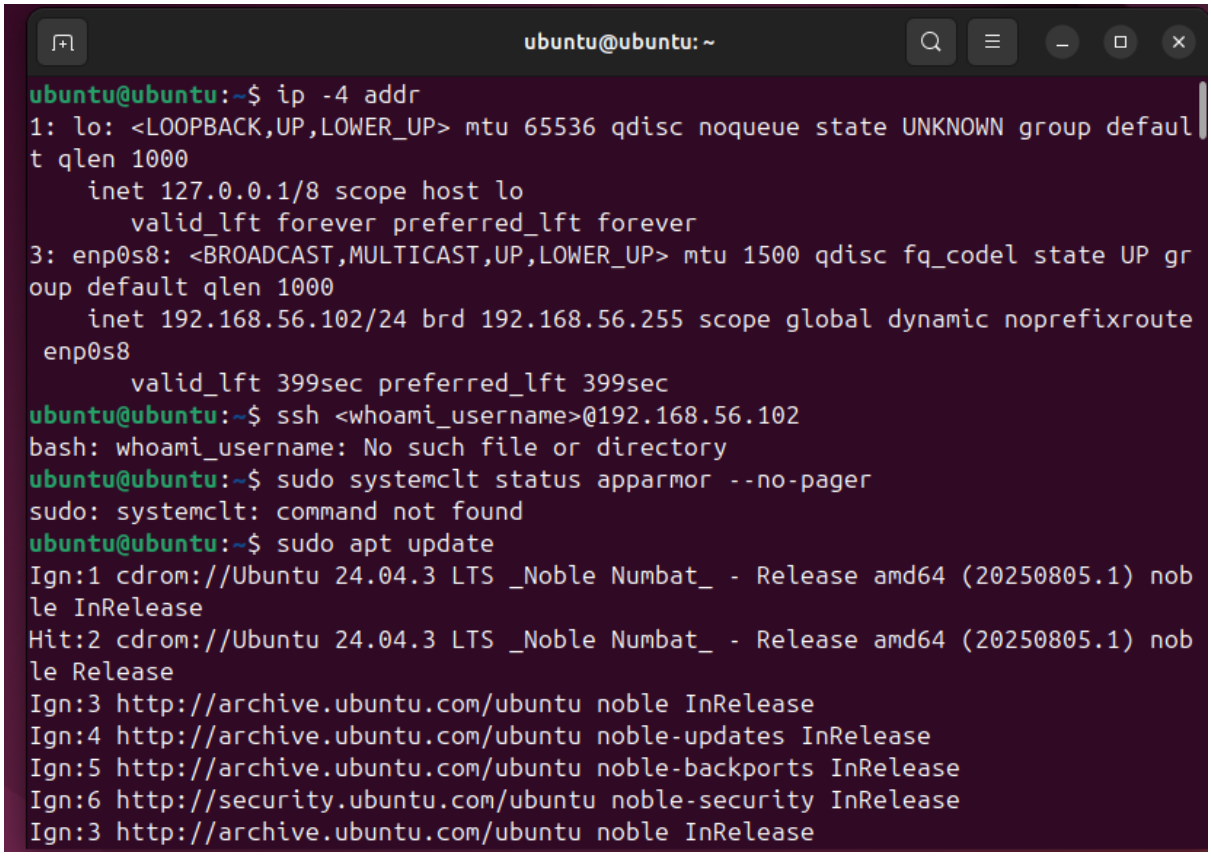
Writing and executing scripts for security verification and monitoring reinforced the importance of automation in professional system administration. This week significantly improved my understanding of how secure systems are maintained and monitored in real-world environments.


ubuntu@ubuntu: ~
0 processes are in mixed mode.
ubuntu@ubuntu:~$ sudo apt -y intall unattended-upgrades
E: Invalid operation intall
ubuntu@ubuntu:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades
ubuntu@ubuntu:~$ cat/etc/apt.conf.d/20auto-upgrades
bash: cat/etc/apt.conf.d/20auto-upgrades: No such file or directory
ubuntu@ubuntu:~$ sudo systemctl status unattended-upgrades --no-pager
● unattended-upgrades.service - Unattended Upgrades Shutdown
Loaded: loaded (/usr/lib/systemd/system/unattended-upgrades.service; enable
d; preset: enabled)
Active: active (running) since Wed 2025-12-17 16:55:01 UTC; 30min ago
Docs: man:unattended-upgrade(8)
Main PID: 1877 (unattended-upgr)
Tasks: 2 (limit: 2254)
Memory: 9.8M (peak: 11.6M)
CPU: 516ms
CGroup: /system.slice/unattended-upgrades.service
└1877 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-…
Dec 17 16:55:01 ubuntu systemd[1]: Started unattended-upgrades.service - Un…own.
Hint: Some lines were ellipsized, use -l to show in full.
ubuntu@ubuntu:~$ sudo apt -y install fail2ban
Reading package lists... Done
Building dependency tree... Done


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sudo aa-status
apparmor module is loaded.
35 profiles are loaded.
30 profiles are in enforce mode.
/snap/snapd/24792/usr/lib/snapd/snap-confine
/snap/snapd/24792/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/lib/snapd/snap-confine
/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
rsyslogd
snap-update-ns.firefox
snap-update-ns.firmware-updater
snap-update-ns.snap-store
snap-update-ns.snapd-desktop-integration
snap-update-ns.thunderbird
snap-update-ns.ubuntu-desktop-bootstrap
snap.firefox.firefox
snap.firefox.geckodriver
snap.firefox.hook.configure
snap.firefox.hook.disconnect-plug-host-hunspell
snap.firefox.hook.install
snap.firefox.hook.post-refresh
snap.firmware-updater.firmware-notifier
snap.firmware-updater.firmware-updater
snap.firmware-updater.firmware-updater-app

```
ubuntu@ubuntu:~$ sudo apt -y install apparmor-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-apparmor python3-libapparmor
Suggested packages:
  vim-addon-manager
The following NEW packages will be installed:
  apparmor-utils python3-apparmor python3-libapparmor
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 161 kB of archives.
After this operation, 1040 kB of additional disk space will be used.
Ign:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-libappar
mor amd64 4.0.1really4.0.1-0ubuntu0.24.04.4
Ign:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-apparmor
 all 4.0.1really4.0.1-0ubuntu0.24.04.4
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apparmor-utils a
ll 4.0.1really4.0.1-0ubuntu0.24.04.4
Ign:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-libappar
mor amd64 4.0.1really4.0.1-0ubuntu0.24.04.4
Ign:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-apparmor
 all 4.0.1really4.0.1-0ubuntu0.24.04.4
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apparmor-utils a
```

```
ubuntu@ubuntu:~$ sudo apt update
Ign:1 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) nob
le InRelease
Hit:2 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) nob
le Release
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Err:6 http://security.ubuntu.com/ubuntu noble-security InRelease
  Temporary failure resolving 'security.ubuntu.com'
Err:3 http://archive.ubuntu.com/ubuntu noble InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
```


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ip -4 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8
valid_lft 399sec preferred_lft 399sec
ubuntu@ubuntu:~$ ssh <whoami_username>@192.168.56.102
bash: whoami_username: No such file or directory
ubuntu@ubuntu:~$ sudo systemclt status apparmor --no-pager
sudo: systemclt: command not found
ubuntu@ubuntu:~$ sudo apt update
Ign:1 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) noble InRelease
Hit:2 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) noble Release
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease