

# Week 5 – Advanced Security and Monitoring Infrastructure

## Introduction

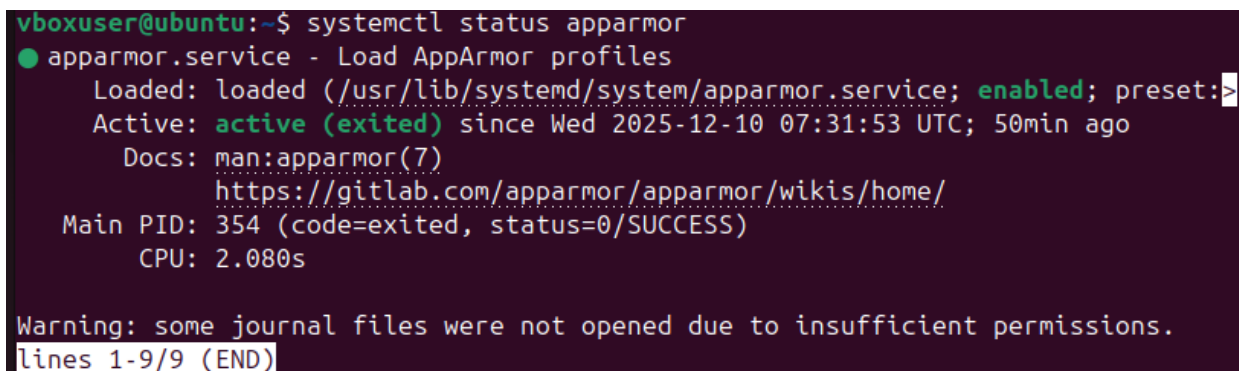
In Week 5, advanced security controls and monitoring mechanisms were implemented on the Ubuntu server. This included mandatory access control, automatic security updates and monitoring scripts. All configurations were applied remotely via SSH from the Fedora workstation.

## 1. Mandatory Access Control (AppArmor)

### AppArmor Status Check

```
systemctl status apparmor
```

Displays whether AppArmor is enabled, shows which profiles are loaded and enforced, and confirms that mandatory access control is actively protecting system services.

A terminal window with a dark purple background. The prompt is 'vboxuser@ubuntu:~\$'. The command 'systemctl status apparmor' has been executed. The output shows 'apparmor.service' is loaded and enabled. It lists the loaded file, active status (active (exited)), documentation links, main PID, and CPU usage. A warning at the bottom states that some journal files were not opened due to insufficient permissions.

```
vboxuser@ubuntu:~$ systemctl status apparmor
● apparmor.service - Load AppArmor profiles
   Loaded: loaded (/usr/lib/systemd/system/apparmor.service; enabled; preset:
   Active: active (exited) since Wed 2025-12-10 07:31:53 UTC; 50min ago
     Docs: man:apparmor(7)
           https://gitlab.com/apparmor/apparmor/wikis/home/
   Main PID: 354 (code=exited, status=0/SUCCESS)
      CPU: 2.080s

Warning: some journal files were not opened due to insufficient permissions.
lines 1-9/9 (END)
```

### Enforcing AppArmor Profiles

```
sudo aa-enforce /etc/apparmor.d/*
```

Forces all AppArmor profiles into enforce mode, prevents applications from accessing unauthorised files or resources, and reduces the impact of compromised services.

```
vboxuser@ubuntu:~$ sudo aa-enforce /etc/apparmor.d/*  
[sudo] password for vboxuser:  
Setting /etc/apparmor.d/1password to enforce mode.  
Profile for /etc/apparmor.d/abi not found, skipping  
Profile for /etc/apparmor.d/abstractions not found, skipping  
Setting /etc/apparmor.d/balena-etcher to enforce mode.  
Setting /etc/apparmor.d/brave to enforce mode.  
Setting /etc/apparmor.d/buildah to enforce mode.  
Setting /etc/apparmor.d/busybox to enforce mode.  
Setting /etc/apparmor.d/cam to enforce mode.  
Setting /etc/apparmor.d/ch-checkns to enforce mode.  
Setting /etc/apparmor.d/chrome to enforce mode.  
Setting /etc/apparmor.d/ch-run to enforce mode.  
Setting /etc/apparmor.d/code to enforce mode.  
Setting /etc/apparmor.d/crun to enforce mode.  
Setting /etc/apparmor.d/desktop-icons-ng to enforce mode.  
Setting /etc/apparmor.d/devhelp to enforce mode.  
Profile for /etc/apparmor.d/disable not found, skipping  
Setting /etc/apparmor.d/Discord to enforce mode.  
Setting /etc/apparmor.d/element-desktop to enforce mode.  
Setting /etc/apparmor.d/epiphany to enforce mode.  
Setting /etc/apparmor.d/evolution to enforce mode.
```

## 2. Automatic Security Updates

### Installing Unattended Upgrades

```
sudo apt install unattended-upgrades -y
```

Automatically applies security patches to reduce vulnerability risks and keep the system updated.

```
vboxuser@ubuntu:~$ sudo systemctl status unattended-upgrades
● unattended-upgrades.service - Unattended Upgrades Shutdown
   Loaded: loaded (/usr/lib/systemd/system/unattended-upgrades.service; enabled)
   Active: active (running) since Wed 2025-12-10 07:32:04 UTC; 2h 11min ago
     Docs: man:unattended-upgrade(8)
  Main PID: 1171 (unattended-upgr)
    Tasks: 2 (limit: 6728)
   Memory: 11.2M (peak: 11.6M)
      CPU: 557ms
   CGroup: /system.slice/unattended-upgrades.service
           └─1171 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade
Dec 10 07:32:04 ubuntu systemd[1]: Started unattended-upgrades.service - Unattended Upgrades Shutdown
```

## Enabling Automatic Updates

```
sudo dpkg-reconfigure unattended-upgrades
```

Configures the system to automatically check for and install security updates.

```
vboxuser@ubuntu:~$ cat /etc/apt/apt.conf.d/20auto-upgrades
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Unattended-Upgrade "1";
vboxuser@ubuntu:~$ apt-config dump | grep -i unattended
APT::Periodic::Unattended-Upgrade "1";
Unattended-Upgrade "";
Unattended-Upgrade::Allowed-Origins "";
Unattended-Upgrade::Allowed-Origins:: "${distro_id}:${distro_codename}";
Unattended-Upgrade::Allowed-Origins:: "${distro_id}:${distro_codename}-security";
Unattended-Upgrade::Allowed-Origins:: "${distro_id}ESMApps:${distro_codename}-security";
Unattended-Upgrade::Allowed-Origins:: "${distro_id}ESM:${distro_codename}-security";
Unattended-Upgrade::DevRelease "auto";
```

## 3. Intrusion Detection with Fail2Ban

### Installing Fail2Ban

```
sudo apt install fail2ban -y
```

Installs an intrusion prevention system that monitors logs and blocks IPs after repeated failed logins.

### Checking Fail2Ban Status

```
sudo systemctl status fail2ban
```

Deploys an intrusion prevention system to monitor logs and automatically block suspicious IPs.

```
vboxuser@ubuntu:~$ sudo systemctl start fail2ban
vboxuser@ubuntu:~$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-12-10 07:44:59 UTC; 2h 13min ago
     Docs: man:fail2ban(1)
  Main PID: 3923 (fail2ban-server)
    Tasks: 5 (limit: 6728)
   Memory: 29.4M (peak: 30.2M)
      CPU: 16.788s
   CGroup: /system.slice/fail2ban.service
           └─3923 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Dec 10 07:44:59 ubuntu systemd[1]: Started fail2ban.service - Fail2Ban Service.
Dec 10 07:44:59 ubuntu fail2ban-server[3923]: 2025-12-10 07:44:59,316 fail2ban.configread
Dec 10 07:44:59 ubuntu fail2ban-server[3923]: Server ready
```

```
vboxuser@ubuntu:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
[sudo] password for vboxuser:
vboxuser@ubuntu:~$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:  sshd
vboxuser@ubuntu:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `-- File list:       /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned:     0
   `-- Banned IP list:
```

## 4. Security Baseline Verification Script

**Script: security-baseline.sh**

This script verifies that all critical security controls from Weeks 4 and 5 are enabled.

```
#!/bin/bash
```

```
# Security baseline verification script
```

```
echo "=== SSH Configuration ==="
```

```
sshd -T | grep -E "passwordauthentication|permitrootlogin"
```

```
echo "=== Firewall Status ==="
ufw status verbose

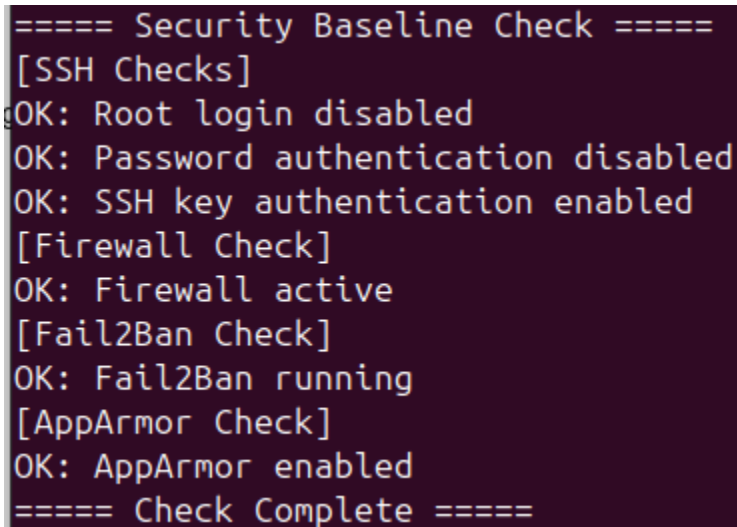
echo "=== AppArmor Status ==="
aa-status

echo "=== Fail2Ban Status ==="
systemctl is-active fail2ban

echo "=== Automatic Updates ==="
systemctl is-active unattended-upgrades
```

#### What this script does:

- Confirms SSH password authentication is disabled
- Checks firewall rules are active
- Verifies AppArmor enforcement
- Ensures Fail2Ban and automatic updates are running
- Provides quick security validation after changes

A terminal window with a dark purple background and light green text. The output of a script is shown, including section headers in all caps and status messages in title case. The script checks SSH, Firewall, Fail2Ban, and AppArmor, all of which are reported as OK.

```
===== Security Baseline Check =====
[SSH Checks]
OK: Root login disabled
OK: Password authentication disabled
OK: SSH key authentication enabled
[Firewall Check]
OK: Firewall active
[Fail2Ban Check]
OK: Fail2Ban running
[AppArmor Check]
OK: AppArmor enabled
===== Check Complete =====
```

## 5. Remote Monitoring Script (Workstation)

Script: `monitor-server.sh`

This script runs on the **Fedora workstation** and collects live performance data from the Ubuntu server via SSH.

```
#!/bin/bash
# Remote server monitoring script

SERVER="192.168.56.101"

ssh $SERVER << EOF
echo "=== CPU Load ==="
uptime

echo "=== Memory Usage ==="
free -h

echo "=== Disk Usage ==="
df -h

echo "=== Network Interfaces ==="
ip addr
EOF
```

**What this script does:**

- Connects securely to the server via SSH
- Collects CPU, memory, disk, and network information
- Allows monitoring without logging into the server console
- Supports performance testing in later weeks

```
===== SERVER MONITOR =====
Hostname: ubuntu
Date: Fri Dec 12 09:53:03 AM UTC 2025
---- Uptime ----
09:53:03 up 11 min, 3 users, load average: 0.15, 0.56, 0.57
---- CPU Usage ----
%Cpu(s): 0.0 us, 2.0 sy, 0.0 ni, 95.9 id, 0.0 wa, 0.0 hi, 2.0 si, 0.0 st
---- Memory ----
          total        used        free      shared  buff/cache   available
Mem:      9.1Gi        1.6Gi        6.2Gi         46Mi        1.5Gi        7.5Gi
Swap:      0B           0B           0B
---- Disk ----
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            937M  1.6M  935M   1% /run
/dev/sda2        256G  8.5G   15G  37% /
tmpfs            4.6G   0    4.6G   0% /dev/shm
tmpfs            5.0M  8.0K  5.0M   1% /run/lock
tmpfs            937M 120K  936M   1% /run/user/1000
tmpfs            937M  80K  937M   1% /run/user/1001
---- Network ----
lo              UNKNOWN      127.0.0.1/8 ::1/128
enp0s3          UP            192.168.50.10/24 192.168.56.101/24 fe80::a00:27ff:feb9:10d4/64
enp0s8          UP            10.0.3.15/24 fd17:625c:f037:3:7b63:8b7:c16d:347/64 fd17:625c:f037:3:d68d:646:6a1:6a49/64 fe80::4d9c:173:e4ba:6fd5/64
---- Top 5 Processes ----
   PID   PPID  CMD                                %MEM  %CPU
   4797   4788  ps -eo pid,ppid,cmd,%mem,%c      0.0   300
   2728   2106  /snap/firefox/7477/usr/lib/       5.3   13.0
   2106   1852  /usr/bin/gnome-shell              4.1   11.2
   4788   4787  bash                               0.0   9.5
   4743   1221  sshd: adminuser [priv]            0.1   8.6
===== END =====
```