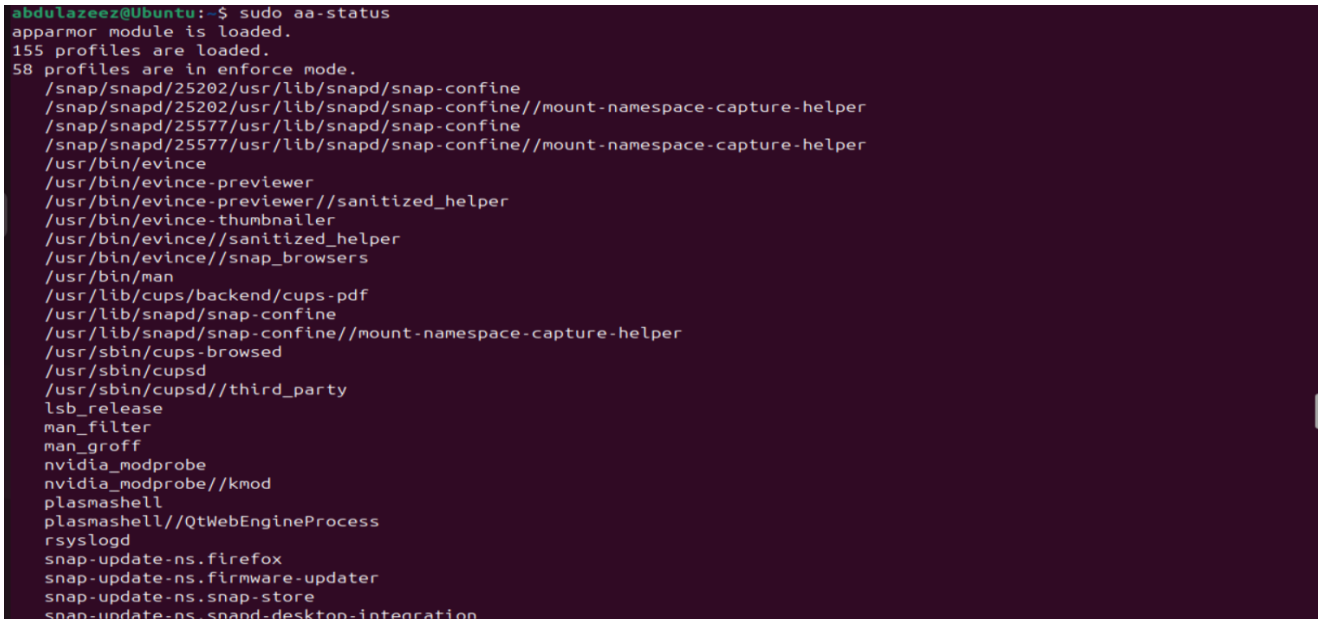
Week 5 Journal: Security Automation & Remote Monitoring

Introduction

This week focused on securing my Ubuntu server and setting up remote monitoring from my Linux Mint workstation. I configured AppArmor, automatic updates, fail2ban, and created two scripts: security-baseline.sh and monitor-server.sh. Below is my Week 5 journal detailing everything I did.

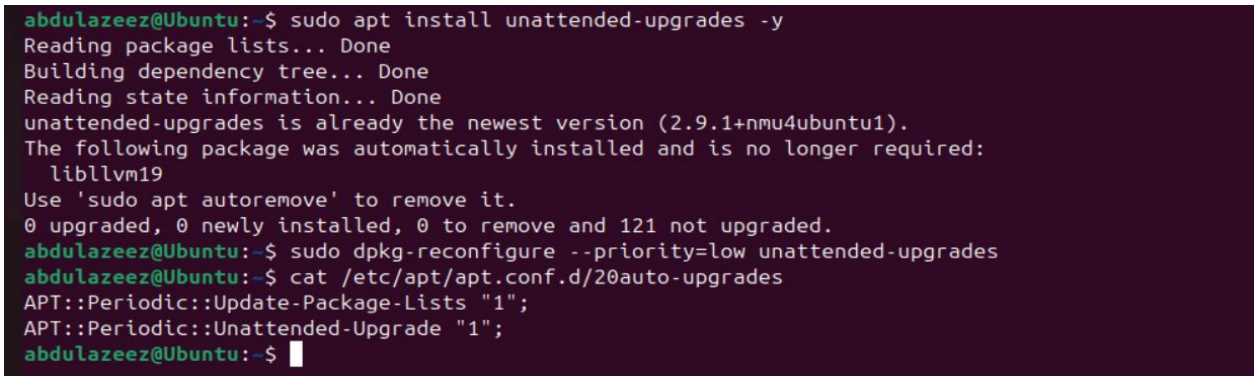
1. AppArmor Access Control

I verified AppArmor was running using `sudo aa-status`. Since Ubuntu did not include a default profile for sshd, I created my own AppArmor profile file and loaded it using apparmor_parser. I switched it to complain mode later so SSH could run normally.



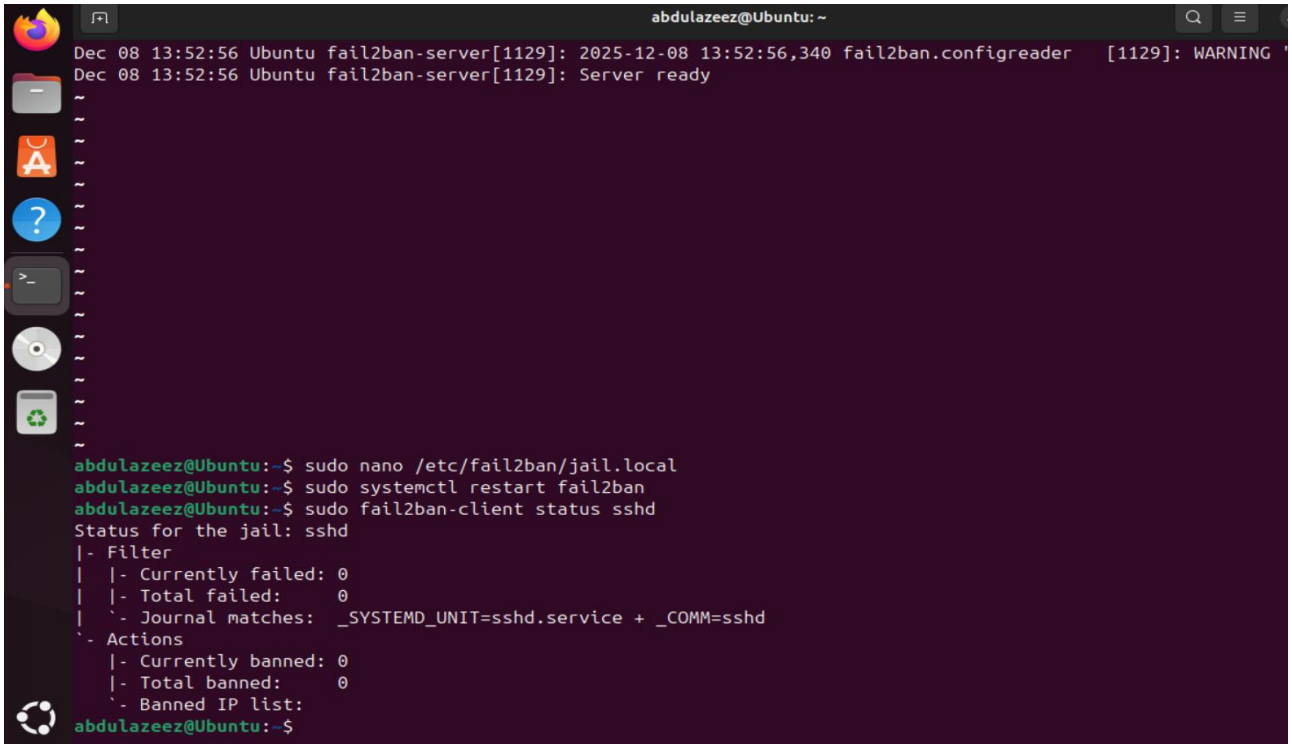
2. Automatic Security Updates

I installed and enabled unattended-upgrades to automate security patching. I checked the configuration using `cat /etc/apt/apt.conf.d/20auto-upgrades`, which confirmed automatic updates were enabled.



3. Fail2ban – SSH Protection

I installed fail2ban and configured the SSH jail in `/etc/fail2ban/jail.local`. After restarting the service, I confirmed the jail was active using `fail2ban-client status sshd`.

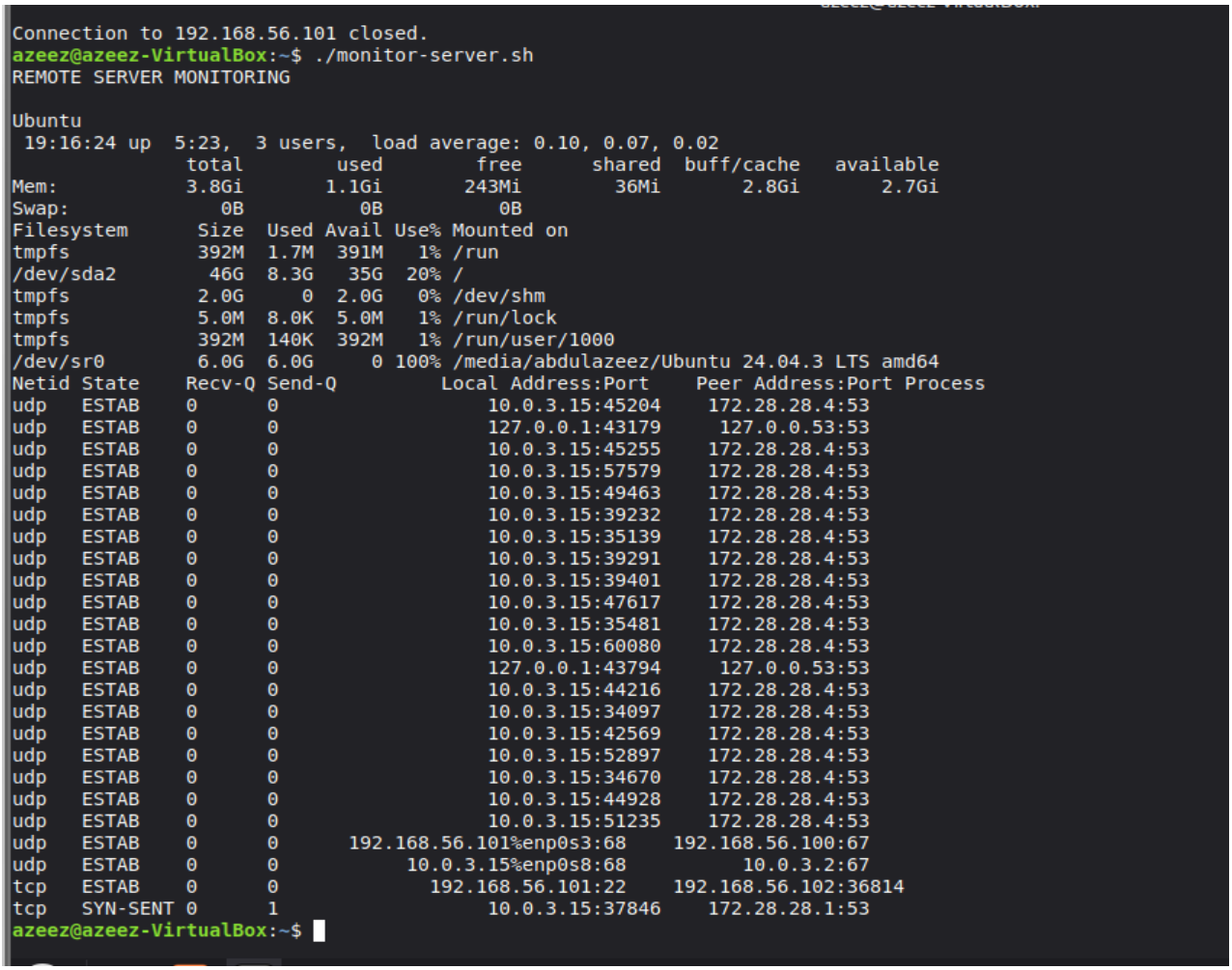


4. security-baseline.sh Script

I wrote a script that prints a full security baseline report including firewall status, AppArmor, SSH config, fail2ban, and automatic update settings.

5. monitor-server.sh – Remote Monitoring

On Linux Mint, I created a monitoring script that connects to Ubuntu through SSH and prints hostname, uptime, memory usage, disk usage, and network connections.



Reflection

This week was challenging, especially dealing with AppArmor profiles and SSH issues, but I learned a lot about real server security. Setting up fail2ban and automatic updates improved my understanding of how to protect Linux systems. The monitoring script also helped me understand remote server management better.