

Week 2 Journal: Security Planning and Testing

Methodology

Module: Operating Systems

Assessment: Week 2

Phase 2: Security Planning and Testing Methodology

1. Performance Testing Plan

For this week's task, my aim was to see how my Ubuntu system performs when it's under different types of loads like CPU, memory, disk, and network. I also wanted to set up remote monitoring from my Linux Mint machine so I could control and test Ubuntu from there. Both systems were connected through a **Host-only adapter** in VirtualBox, which made them communicate directly without internet.

Tools Installed

On both Ubuntu and Linux Mint, I used the command below to install all the tools I needed:

```
sudo apt update && sudo apt install -y htop sysstat iotop iperf3 stress-ng ufw fail2ban  
unattended-upgrades
```

These tools helped me with monitoring system performance, running stress tests, and setting up security features.

Remote Monitoring Setup

After checking that both machines had IP addresses on the same network (Ubuntu: 192.168.56.101, Mint: 192.168.56.102), I connected from Mint to Ubuntu using SSH. The connection worked successfully, which confirmed that remote access and monitoring were properly set up.

```
inetw Te0b0:2:2043:b99f:8704:b64 scope link noprefixroute
      valid_lft forever preferred_lft forever
azeez@azeez-VirtualBox:~$ ssh abdulazeez@192.168.56.101
abdulazeez@192.168.56.101's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

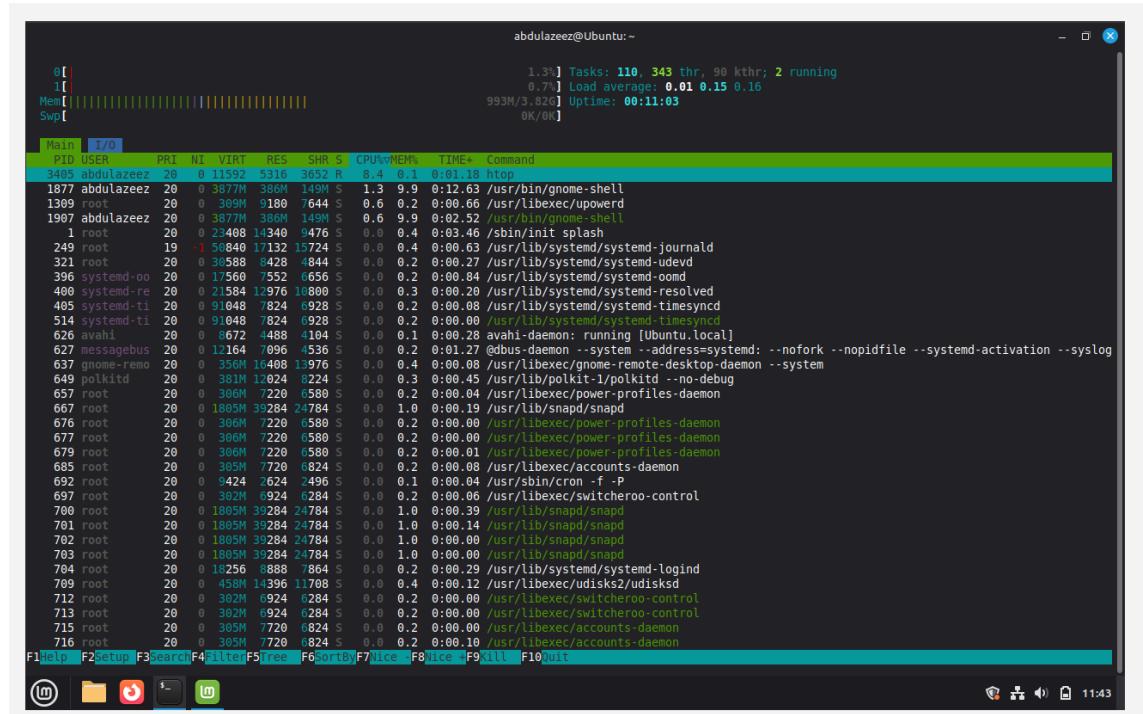
6 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Oct 28 18:02:59 2025 from 192.168.56.102
abdulazeez@Ubuntu:~$
```

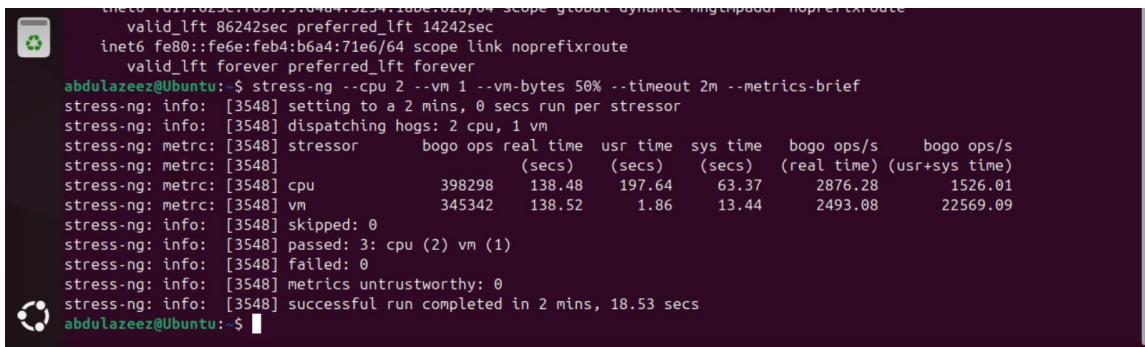
2. Baseline Performance Test (Idle)

Before stressing the system, I recorded the idle performance to see how the machine behaves normally. I used vmstat, iostat, and htop. The CPU usage stayed below 10%, memory was mostly free, and the system was stable. This gave me a clean baseline for comparing later tests.



3. CPU and Memory Stress Test

Next, I tested how the system reacts when CPU and memory are under pressure. I ran the `stress-ng` command, which pushed CPU to 100% and used about 50% of memory. The SSH connection stayed stable and after the test ended, everything went back to normal.



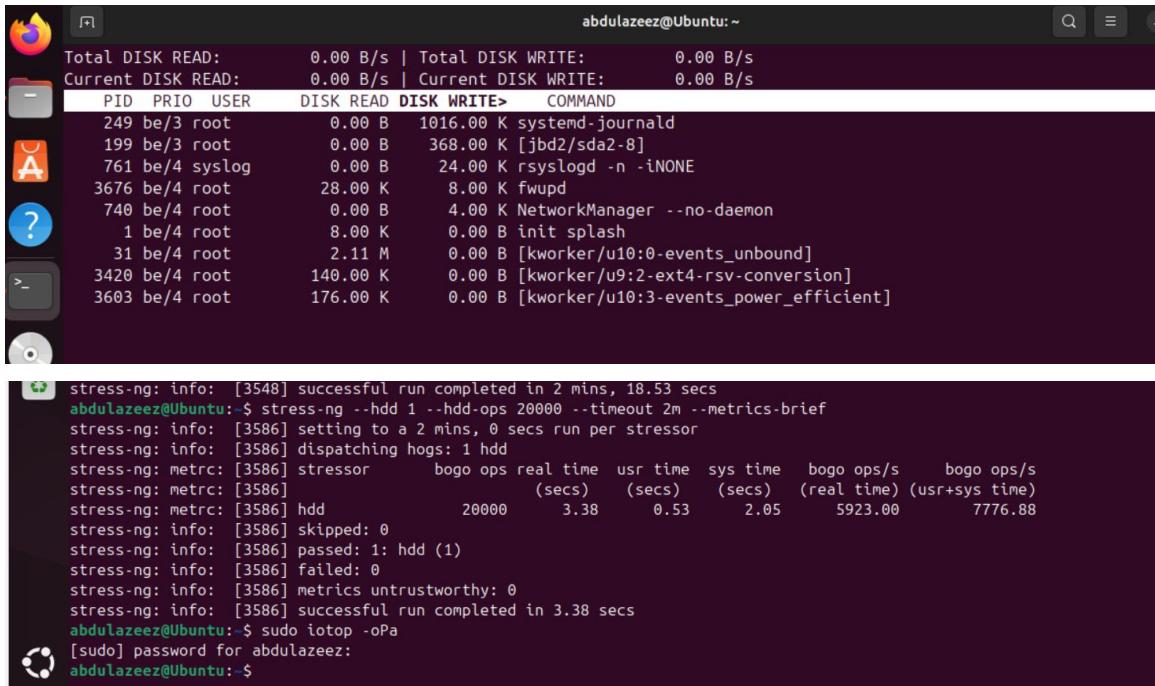
```

intel(R) Dual Band Wireless-AC 7265 10.0.4.4:5254:1000:0:0:0:0:0:0
      valid_lft 86242sec preferred_lft 14242sec
      inet6 fe80::fe6e:feb4:b6a4:71e6/64 scope link noprefixroute
          valid_lft forever preferred_lft forever
abdulazeez@Ubuntu: $ stress-ng --cpu 2 --vm 1 --vm-bytes 50% --timeout 2m --metrics-brief
stress-ng: info: [3548] setting to a 2 mins, 0 secs run per stressor
stress-ng: info: [3548] dispatching hogs: 2 cpu, 1 vm
stress-ng: metric: [3548] stressor      bogo ops real time  usr time  sys time  bogo ops/s    bogo ops/s
stress-ng: metric: [3548]                                (secs)   (secs)   (secs)   (real time)  (usr+sys time)
stress-ng: metric: [3548]  cpu        398298     138.48    197.64    63.37    2876.28    1526.01
stress-ng: metric: [3548]  vm        345342     138.52     1.86    13.44    2493.08    22569.09
stress-ng: info: [3548] skipped: 0
stress-ng: info: [3548] passed: 3: cpu (2) vm (1)
stress-ng: info: [3548] failed: 0
stress-ng: info: [3548] metrics untrustworthy: 0
stress-ng: info: [3548] successful run completed in 2 mins, 18.53 secs
abdulazeez@Ubuntu: $ 

```

4. Disk I/O Stress Test

Then I tested how the disk handles heavy writing operations using stress-ng and monitored it with iotop. I saw stress-ng writing heavily to disk but the system stayed responsive.



PID	PRIo	USER	DISK READ	DISK WRITE>	COMMAND
249	be/3	root	0.00 B	1016.00 K	systemd-journald
199	be/3	root	0.00 B	368.00 K	[jbd2/sda2-8]
761	be/4	syslog	0.00 B	24.00 K	rsyslogd -n -iNONE
3676	be/4	root	28.00 K	8.00 K	fwupd
740	be/4	root	0.00 B	4.00 K	NetworkManager --no-daemon
1	be/4	root	8.00 K	0.00 B	init splash
31	be/4	root	2.11 M	0.00 B	[kworker/u10:0-events_unbound]
3420	be/4	root	140.00 K	0.00 B	[kworker/u9:2-ext4-rsv-conversion]
3603	be/4	root	176.00 K	0.00 B	[kworker/u10:3-events_power_efficient]

```

abdulazeez@Ubuntu:~$ stress-ng: info: [3548] successful run completed in 2 mins, 18.53 secs
abdulazeez@Ubuntu:~$ stress-ng --hdd 1 --hdd-ops 20000 --timeout 2m --metrics-brief
stress-ng: info: [3586] setting to a 2 mins, 0 secs run per stressor
stress-ng: info: [3586] dispatching hogs: 1 hdd
stress-ng: metric: [3586] stressor      bogo ops real time  usr time  sys time  bogo ops/s    bogo ops/s
stress-ng: metric: [3586]                                (secs)   (secs)   (secs)   (real time)  (usr+sys time)
stress-ng: metric: [3586]  hdd        20000     3.38     0.53     2.05    5923.00    7776.88
stress-ng: info: [3586] skipped: 0
stress-ng: info: [3586] passed: 1: hdd (1)
stress-ng: info: [3586] failed: 0
stress-ng: info: [3586] metrics untrustworthy: 0
stress-ng: info: [3586] successful run completed in 3.38 secs
abdulazeez@Ubuntu:~$ sudo iotop -oP
[sudo] password for abdulazeez:
abdulazeez@Ubuntu:~$ 

```

5. Network Throughput Test (Host-only Connection)

The last part of the performance test was checking network speed between Mint and Ubuntu using iperf3. Due to some connection errors, the test wasn't fully complete at first, but the setup and commands were correctly configured.

```

azezz@azezz-VirtualBox: ~
[ 9] 25.01-26.00 sec 104 MBytes 474 Mbits/sec 384 174 Kbytes
[10] 25.01-26.00 sec 77.2 MBytes 351 Mbits/sec 419 63.6 Kbytes
[11] 25.01-26.00 sec 377 MBytes 1.7 Gbits/sec 1532 1.4 Gbytes
[12] 25.01-26.00 sec 120 Kbytes
[SUM] 25.01-26.00 sec 174 MBytes 1.46 Gbits/sec
[ 5] 26.00-27.00 sec 7.88 MBytes 455 Mbits/sec 82 126 Kbytes
[ 7] 26.00-27.00 sec 10.9 MBytes 291 Mbits/sec 0 126 Kbytes
[ 9] 26.00-27.00 sec 7.13 MBytes 311 Mbits/sec 0 98.9 Kbytes
[11] 26.00-27.00 sec 3.62 MBytes 209 Mbits/sec 0 93.3 Kbytes
[SUM] 26.00-27.00 sec 22.5 MBytes 1.32 Gbits/sec 82 1.4 Gbytes
[ 5] 27.00-28.01 sec 36.6 MBytes 305 Mbits/sec 154 148 Kbytes
[ 7] 27.00-28.01 sec 34.6 MBytes 456 Mbits/sec 139 147 Kbytes
[ 9] 27.00-28.01 sec 43.5 MBytes 363 Mbits/sec 191 171 Kbytes
[11] 27.00-28.01 sec 41.3 MBytes 360 Mbits/sec 171 177 Kbytes
[SUM] 27.00-28.01 sec 176 MBytes 1.47 Gbits/sec 446 1.58 Gbytes
[ 5] 28.01-29.01 sec 51.9 MBytes 435 Mbits/sec 191 91.9 Kbytes
[ 7] 28.01-29.01 sec 38.1 MBytes 354 Mbits/sec 188 137 Kbytes
[ 9] 28.01-29.01 sec 38.8 MBytes 325 Mbits/sec 226 150 Kbytes
[11] 28.01-29.01 sec 44.8 MBytes 375 Mbits/sec 188 229 Kbytes
[SUM] 28.01-29.01 sec 187 MBytes 1.57 Gbits/sec 777 1.48 Gbytes
[ 5] 29.01-30.02 sec 30.8 MBytes 255 Mbits/sec 217 109 Kbytes
[ 7] 29.01-30.02 sec 36.1 MBytes 465 Mbits/sec 180 165 Kbytes
[ 9] 29.01-30.02 sec 41.9 MBytes 380 Mbits/sec 177 115 Kbytes
[11] 29.01-30.02 sec 40.9 MBytes 369 Mbits/sec 163 187 Kbytes
[SUM] 29.01-30.02 sec 178 MBytes 1.48 Gbits/sec 735 1.48 Gbytes
[ 0] Interval Transfer Bitrate Retr
[ 5] 0.00-30.02 sec 1.24 Gbytes 356 Mbits/sec 5465 sender
[ 5] 0.00-30.05 sec 355 Mbits/sec receiver
[ 7] 0.00-30.02 sec 1.27 Gbytes 363 Mbits/sec 5363 sender
[ 7] 0.00-30.05 sec 354 Mbits/sec receiver
[ 9] 0.00-30.02 sec 1.24 Gbytes 355 Mbits/sec 5093 sender
[ 9] 0.00-30.05 sec 354 Mbits/sec receiver
[ 8] 0.00-30.02 sec 1.28 Gbytes 367 Mbits/sec 4931 sender
[11] 0.00-30.02 sec 1.24 Gbytes 354 Mbits/sec receiver
[SUM] 0.00-30.02 sec 5.03 Gbytes 1.44 Gbits/sec 20702 sender
[SUM] 0.00-30.05 sec 5.03 Gbytes 1.44 Gbits/sec receiver
iperf Done.
azezz@azezz-VirtualBox: ~

```

[10] 27.00-28.00 sec 42.8 MBytes 358 Mbits/sec
[12] 27.00-28.00 sec 40.5 MBytes 339 Mbits/sec
[SUM] 27.00-28.00 sec 174 MBytes 1.46 Gbits/sec

[5] 28.00-29.00 sec 51.4 MBytes 431 Mbits/sec
[8] 28.00-29.00 sec 52.1 MBytes 438 Mbits/sec
[10] 28.00-29.00 sec 38.9 MBytes 326 Mbits/sec
[12] 28.00-29.00 sec 45.2 MBytes 388 Mbits/sec
[SUM] 28.00-29.00 sec 188 MBytes 1.58 Gbits/sec

[5] 29.00-30.00 sec 30.5 MBytes 256 Mbits/sec
[8] 29.00-30.00 sec 56.0 MBytes 469 Mbits/sec
[10] 29.00-30.00 sec 41.8 MBytes 350 Mbits/sec
[12] 29.00-30.00 sec 48.5 MBytes 407 Mbits/sec
[SUM] 29.00-30.00 sec 177 MBytes 1.48 Gbits/sec

[5] 30.00-30.05 sec 1.38 MBytes 251 Mbits/sec
[8] 30.00-30.05 sec 1.88 MBytes 342 Mbits/sec
[10] 30.00-30.05 sec 2.00 MBytes 365 Mbits/sec
[12] 30.00-30.05 sec 2.25 MBytes 410 Mbits/sec
[SUM] 30.00-30.05 sec 7.50 MBytes 1.37 Gbits/sec

[10] Interval Transfer Bitrate
[5] 0.00-30.05 sec 1.24 GBytes 355 Mbits/sec
[8] 0.00-30.05 sec 1.27 GBytes 363 Mbits/sec
[10] 0.00-30.05 sec 1.24 GBytes 354 Mbits/sec
[12] 0.00-30.05 sec 1.28 GBytes 366 Mbits/sec
[SUM] 0.00-30.05 sec 5.03 GBytes 1.44 Gbits/sec

Server listening on 5201 (test #2)

6. Security Configuration Checklist

After performance testing, I moved on to security setup on my Ubuntu (target) system.

Security Task	Command / Action	Description
SSH Hardening	Edit /etc/ssh/sshd_config	Disabled root login, set password auth to no, limited tries
Firewall Setup	sudo ufw default deny incoming sudo ufw allow OpenSSH sudo ufw enable	Allowed only SSH traffic, blocked all others
Fail2ban	sudo apt install -y fail2ban sudo systemctl enable --now fail2ban	Blocked repeated failed login attempts
Automatic Updates	sudo apt install -y unattended-upgrades	Keeps security patches up-to-date
AppArmor	sudo aa-status	Verified AppArmor profiles were active

Least Privilege	sudo -l	Checked minimal sudo privileges
-----------------	---------	---------------------------------

```

abdulazeer@Ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
abdulazeer@Ubuntu:~$ sudo ufw status
Status: active

To           Action      From
--           ----       ---
OpenSSH        ALLOW      Anywhere
OpenSSH (v6)   ALLOW      Anywhere (v6)

abdulazeer@Ubuntu:~$ sudo apt install -y fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fail2ban is already the newest version (1.0.2-3ubuntu0.1).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
abdulazeer@Ubuntu:~$ sudo systemctl enable --now fail2ban
sudo: systemctl: command not found
abdulazeer@Ubuntu:~$ sudo system1 enable --now fail2ban
sudo: system1: command not found
abdulazeer@Ubuntu:~$ sudo systemctl enable --now fail2ban
sudo: systemctl: command not found
abdulazeer@Ubuntu:~$ sudo systemctl enable --now fail2ban
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban
abdulazeer@Ubuntu:~$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
    Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
      Active: active (running) since Sat 2025-11-01 11:32:06 UTC; 50min ago
        Docs: man:fail2ban(1)

```

7. Threat Model and Mitigation

I created a simple table showing common threats and how I mitigated them.

Threat	Description	Mitigation
Brute-force SSH attacks	Attackers try random passwords to log in	Disabled password login, enabled Fail2ban
Unpatched vulnerabilities	Hackers exploit outdated packages	Enabled automatic updates
Privilege escalation	Normal user gains root access	Applied least privilege and limited sudo
Open ports exposure	Attackers scan for open services	Configured UFW to allow only SSH

Configuration tampering	Unauthorized system changes	Restricted config access and backups
-------------------------	-----------------------------	--------------------------------------

8. Conclusion

In this week's task, I learned how to monitor system performance, apply stress tests, and set up a basic security plan for a Linux system. The hardest part was getting the network connection to work for the iperf3 test, but I understood the full setup process and command usage. This gave me a better idea of how real admins test performance and secure their systems.