<table>
<tr><td>College Name</td><td colspan="3">University of Hertfordshire SEGi College Subang Jaya</td></tr>
<tr><td>Programme Name</td><td colspan="3">BACHELOR OF SCIENCE (HONS) COMPUTER SCIENCE (CYBER SECURITY AND NETWORKS)</td></tr>
<tr><td rowspan="2">Module Name</td><td rowspan="2">CYBER SECURITY AND NETWORKS PROJECT</td><td>Module Code</td><td>6COM1040</td></tr>
<tr><td>Semester</td><td>September 2025</td></tr>
<tr><td>Module Leader</td><td>Dr. Aneshkumar Thangaveloo</td><td rowspan="2">Assessment Type</td><td rowspan="2">Comprehensive Report</td></tr>
<tr><td>Lecturer Name</td><td>Ms. Nur Diana Madinah Binti Ab Hadi</td></tr>
<tr><td>Student’s declaration</td><td colspan="3">I hereby certify that this assignment is my own work and where materials have been used from other resources, they have been properly acknowledged. I also understand I will face the possibility of failing the module if the content of this assignment is plagiarized.
<table><tr><th>No.</th><th>Name</th><th>Student ID</th><th>Signature / Initial</th></tr><tr><td>1</td><td>PATRICK ROGERS</td><td>SCSJ2100424</td><td>PAT</td></tr></table>Date:</td></tr>
<tr><td>Release Date</td><td></td><td>Submission Due Date</td><td></td><td rowspan="2">Marks obtained:</td></tr>
<tr><td>Date Received</td><td></td><td>Student’s work assessed by / date</td><td></td></tr>
</table>

Module Leader's Feedback.

Module Leader's comments / feedback	
Student's comments	

Table of Contents

List of Figures

1.0 Introduction

A2Z Corporation operates an internet-facing service environment that supports both external clients and internal administrative activities, and because the organization relies heavily on online access and interconnected systems, it faces significant cybersecurity threats such as unauthorized access, network reconnaissance, exploitation of public-facing services, brute-force attacks, packet sniffing, and service disruption, all of which can lead to data loss, operational downtime, and reputational impact if not addressed through proper security architecture and defensive controls. To mitigate these risks, the project focuses on designing, implementing, and validating a secure network environment aligned with industry-standard defensive measures, including secure system architecture design incorporating segmentation and layered defence, deployment of UFW firewall, Docker containerization, and Suricata IDS, along with security testing and vulnerability assessment using tools such as Nmap, Suricata logs, and tcpdump to evaluate system robustness and inform recommendations for improving overall security posture. This report documents the system design, implementation procedures, testing outcomes, and resulting improvements, demonstrating a full lifecycle of cyber defence development aligned with A2Z Corporation's security requirements. This section provides a detailed analysis of the system design, explaining how each network component and security control contributes to A2Z Corporation's defence-in-depth strategy.

2.0 System Design and Security Analysis

A2Z Corporation operates an internet-facing environment that hosts essential services required by external clients, internal employees, and administrative staff. Due to its online exposure and interconnected systems, the organization faces multiple cybersecurity risks that must be mitigated through proper architectural design and layered defence methods. The most significant risks include external cyberattacks (such as port scanning, reconnaissance, brute-force attempts, and exploitation of public-facing services), man-in-the-middle attacks, service disruptions, unauthorized access into the internal network, and zero-day vulnerabilities affecting running services such as web servers or containerized applications.

To address these threats, the designed system architecture adopts a segmented, defence-in-depth security model using a DMZ (Demilitarised Zone), host-level firewall, intrusion detection system (IDS), and container isolation. The DMZ hosts the Ubuntu server running the OWASP Juice Shop web application, while the internal LAN contains only administrative systems. A router acts as the boundary between the internal network, DMZ, and the simulated cloud environment. This structure significantly reduces the attack surface by preventing direct access to internal computers while still enabling secure access to needed services.

The architecture diagram (Figure 1) illustrates how the components are organised and secured within the network.

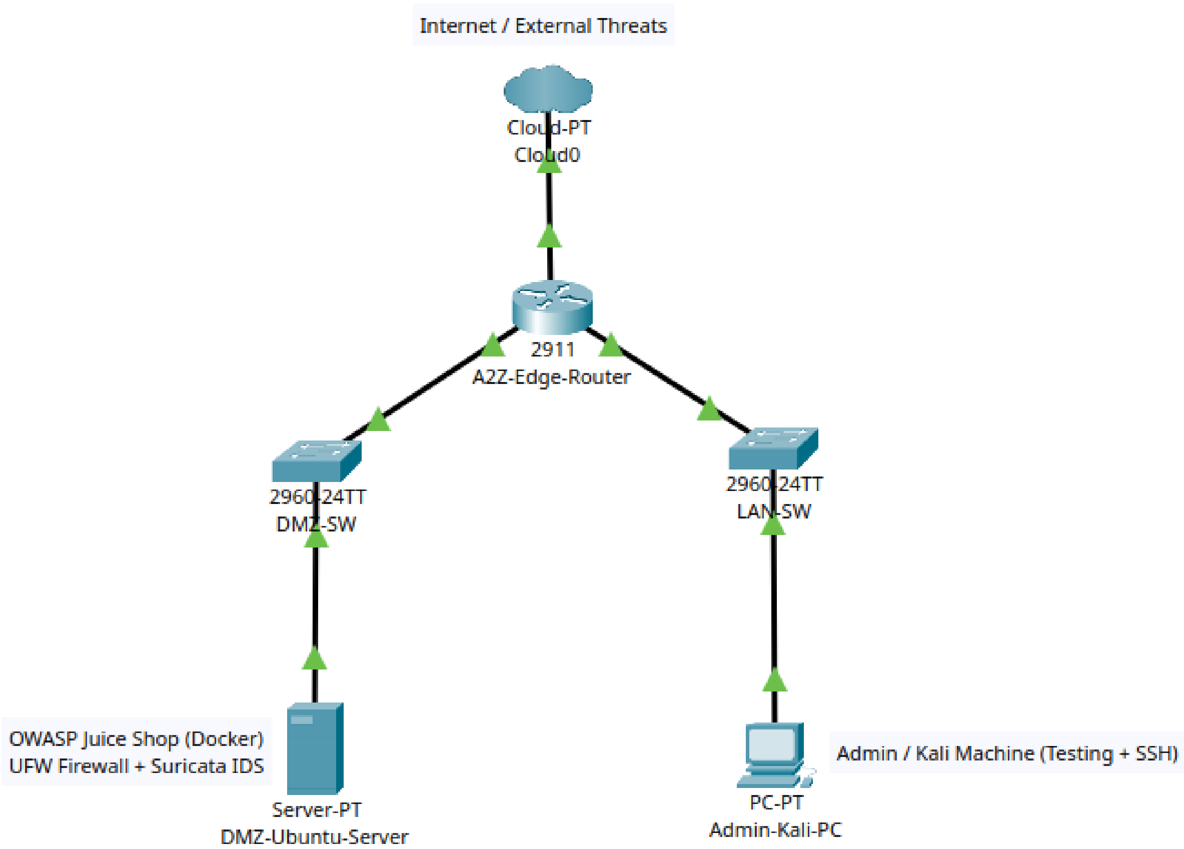


Figure 1: System Architecture Diagram

The diagram shows the following main components:

- **Cloud/Internet** - external environment where potential attackers originate.
- **A2Z-Edge-Router** - isolates and manages traffic flow between DMZ, internal LAN, and cloud.
- **DMZ-SW** - connects the Ubuntu server that hosts the web application.
- **LAN-SW** - connects the admin PC, which performs system management and testing.
- **DMZ-Ubuntu-Server** - hosts Juice Shop using Docker; protected by UFW firewall and monitored by Suricata IDS.
- **Admin-Kali-PC** - used for controlled security testing and administrative access.

This design directly addresses A2Z's most critical security concerns. First, external exposure risk is mitigated by ensuring that only the necessary ports are open on the DMZ server, with all other connections blocked by the firewall. Second, network intrusion and lateral movement risks are minimised by isolating the DMZ from the internal LAN. Even if the public-facing server is compromised, the attacker will not automatically reach internal systems. Third, monitoring and detection risks are handled through Suricata IDS running on the DMZ interface to detect suspicious traffic such as port scans, OS fingerprinting, and brute-force attempts.

Furthermore, containerisation through Docker enhances security by isolating the vulnerable web application from the underlying host operating system. Even if the container is compromised, the attacker does not gain root access to the host machine or the internal network. Encryption measures such as HTTPS are enforced via a Nginx reverse proxy to provide encryption for data in transit, mitigate the risk of interception. The UFW firewall ensures that no unnecessary inbound or outbound communication occurs, providing strict control over external access.

Finally, the system design follows established security principles such as least privilege, segmentation, isolation, and continuous monitoring. These choices align with A2Z Corporation's security needs because they ensure that public services remain available while preventing unrestricted external access, detecting malicious behaviour early, and limiting damage in the event of an attack.

3.0 Implementation of Security Measures

Three primary security measures were implemented: UFW firewall, Docker containerization, and Suricata IDS. These measures were carefully chosen to address A2Z Corporation's critical security risks and provide multiple layers of defence against potential attacks.

3.1 UFW Firewall Configuration

The Ubuntu server's UFW firewall was configured with a default policy set to deny all incoming traffic and allow all outgoing traffic, establishing a basic but solid security posture. This confirms that the firewall is active and enforcing this default policy. Essential services, including SSH (port 22/tcp), HTTP (port 80/tcp), HTTPS (port 443/tcp), and a custom application on port 3000/tcp (Juice Shop), have all been explicitly allowed for incoming connections. A critical finding from the displayed status is that all these ports are allowed from Anywhere for both IPv4 and IPv6 traffic. This configuration protects the system from unauthorized access attempts on non-essential, unused ports by automatically blocking them

```
vboxuser@A2z:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
vboxuser@A2z:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
vboxuser@A2z:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
vboxuser@A2z:~$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
vboxuser@A2z:~$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
vboxuser@A2z:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Figure 2 Firewall Configuration

```
vboxuser@A2z:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
3000/tcp                   ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)
3000/tcp (v6)              ALLOW IN    Anywhere (v6)
```

Figure 3: UFW Firewall Status

3.2 Docker Containerization of Web Application

The OWASP Juice Shop application was deployed using Docker containers, exposing the application on port 3000. Containerization isolates the application from the host operating system and other system services.

```
vboxuser@A2Z:~$ sudo docker ps
CONTAINER ID   IMAGE                  COMMAND                   CREATE
D        STATUS        PORTS                                        NAM
ES
a85f1cc7958c   bkimminich/juice-shop   "/nodejs/bin/node /j…"   2 hour
s ago    Up 2 hours    0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp    har
dcore_panini
606193d7c9b2   nginx:latest            "/docker-entrypoint.…"   2 hour
s ago    Up 2 hours    0.0.0.0:80->80/tcp, [::]:80->80/tcp              web
server
```

Figure 4 Docker Containers Running

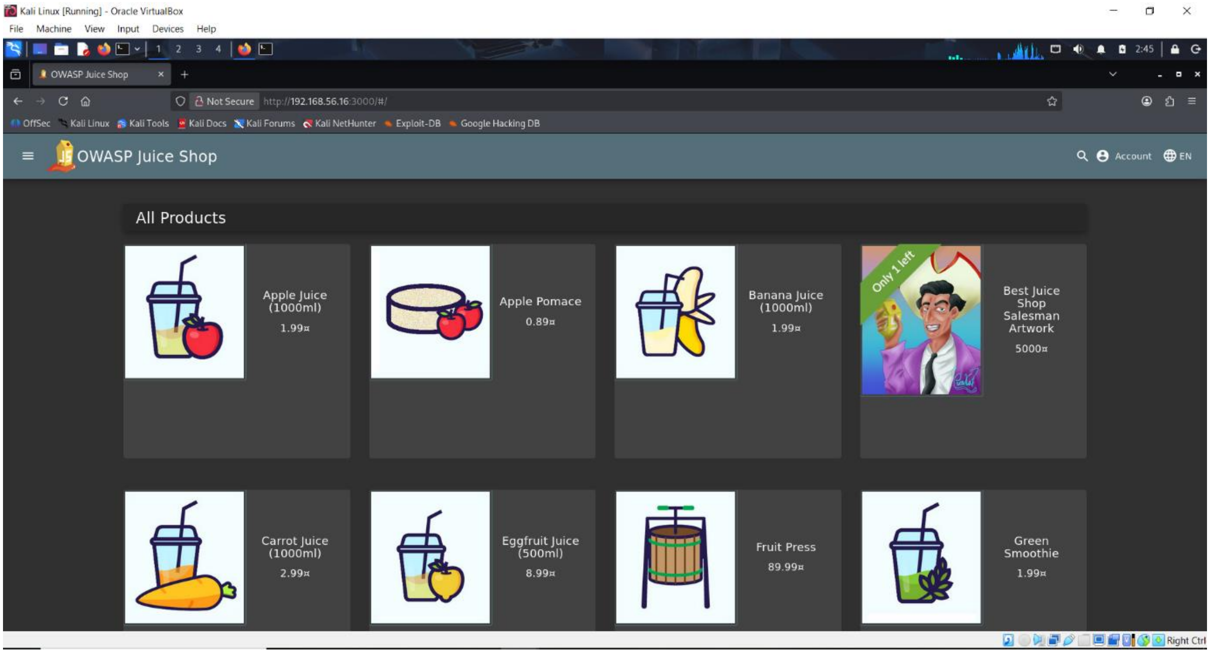


Figure 5 Application Accessible from Browser

Docker enhances security by containing potential breaches within the container environment, preventing attackers from gaining root access to the host server or moving laterally to other systems in the network. If a vulnerability in the web application is exploited, the attacker remains confined to the container, and any malicious changes are erased when the container is restarted. Containerization also simplifies patch management and rollback, allowing administrators to quickly update or replace compromised applications without impacting other services.

3.3 Suricata Intrusion Detection System

Suricata was installed and configured to monitor network traffic on the DMZ interface (enp0s8). Rule sets were loaded successfully, and the IDS engine started in detection mode, analyzing both IP-only and payload-based signatures.

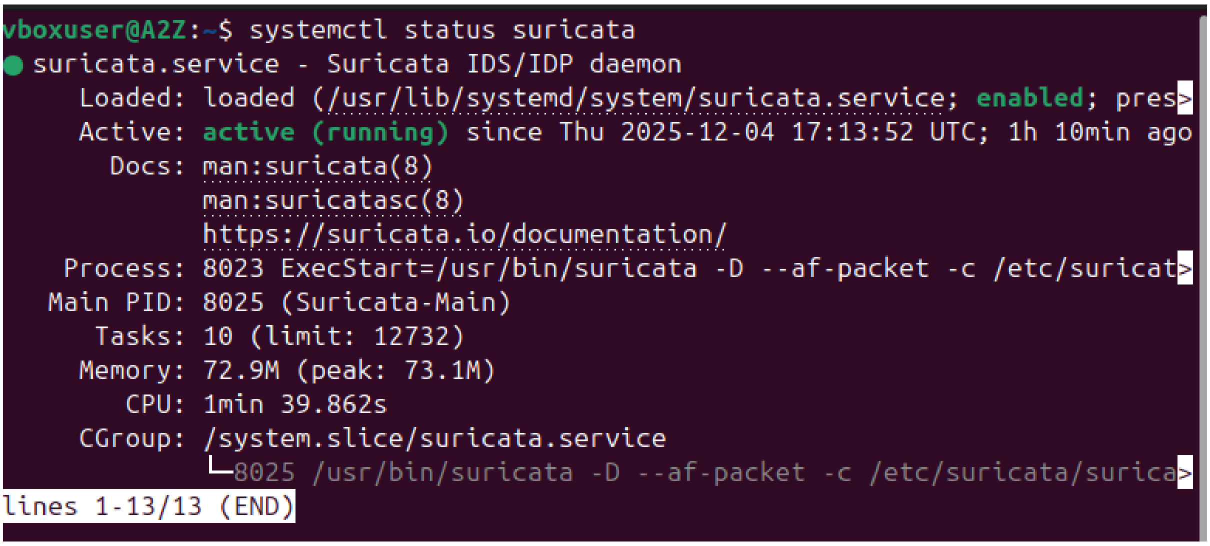


Figure 6 Suricata Running Successfully

Suricata detects real-time tracking, port scanning, operating system fingerprinting, and brute-force attacks. For example, when an attacker runs an aggressive Nmap scan, Suricata quickly detects and logs strange patterns such as SYN flood attempts or OS detection probes. This allows administrators to respond immediately to potential threats, whether by blocking offending IP addresses or reviewing suspicious activity. Suricata thus serves as a proactive security solution, supporting the firewall and container isolation by detecting threats that get past perimeter protections.

By combining these three measures firewall, containerization, and IDS the system implements a defence-in-depth strategy. Each layer addresses specific threat vectors: UFW limits access to essential services, Docker isolates potentially vulnerable applications, and Suricata detects and logs malicious activity for timely intervention.

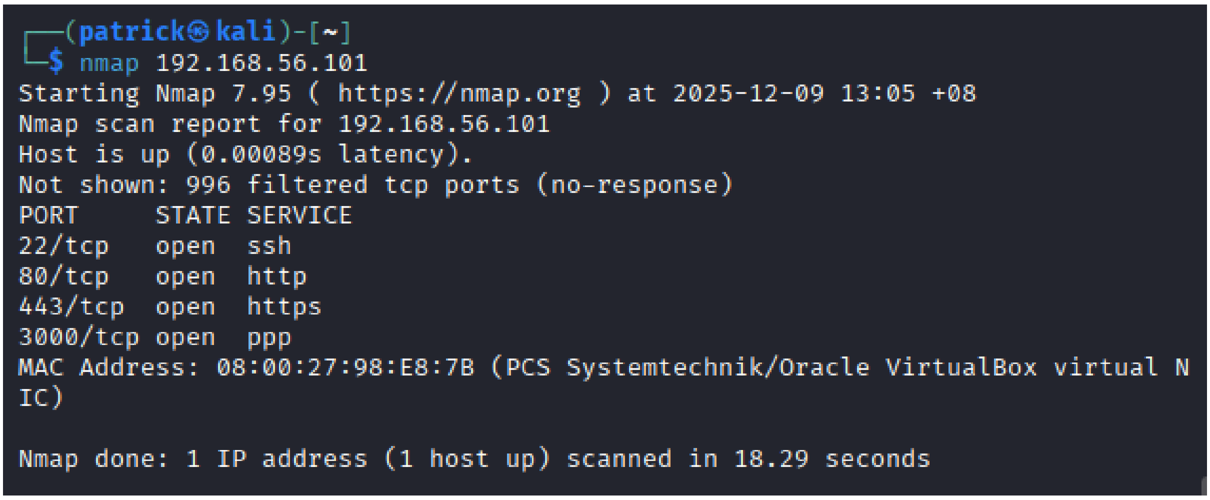
4.0 Testing and Problem-Solving

A thorough testing process was conducted to validate the security measures and identify potential vulnerabilities. Testing was performed from the Kali administrative machine, simulating realistic attacks against the Ubuntu server.

4.1 Nmap Scanning

Nmap is a network scanning tool that detects open ports, running services, and system fingerprints. Several scans were performed:

Basic Scan: Identified open ports (22,80,443,3000).



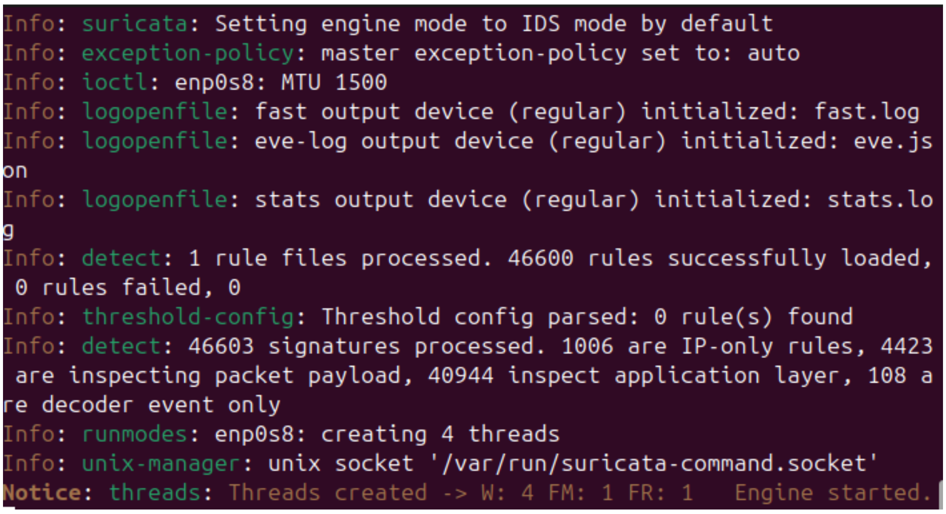
```
┌──(patrick㉿kali)-[~]
└─$ nmap 192.168.56.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 13:05 +08
Nmap scan report for 192.168.56.101
Host is up (0.00089s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3000/tcp open  ppp
MAC Address: 08:00:27:98:E8:7B (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 18.29 seconds
```

Figure 7 Nmap Scan

4.2 Intrusion Detection Logs

Suricata IDS Running Status: To show that Suricata was actively monitoring the DMZ interface, you would use the screenshot that proves the engine started successfully and loaded the detection rules.

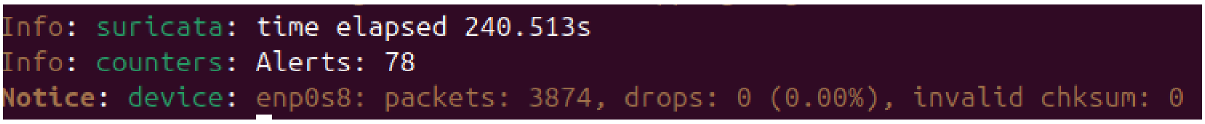


```
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: ioctl: enp0s8: MTU 1500
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.js
on
Info: logopenfile: stats output device (regular) initialized: stats.lo
g
Info: detect: 1 rule files processed. 46600 rules successfully loaded,
 0 rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 46603 signatures processed. 1006 are IP-only rules, 4423
 are inspecting packet payload, 40944 inspect application layer, 108 a
re decoder event only
Info: runmodes: enp0s8: creating 4 threads
Info: unix-manager: unix socket '/var/run/suricata-command.socket'
Notice: threads: Threads created -> W: 4 FM: 1 FR: 1   Engine started.
```

Figure 8 Suricata IDS Running

Confirmed that the Suricata engine was successfully initialized and running in IDS mode, monitoring the DMZ network interface (enp0s8) and processing 46603 detection rules. During testing, Suricata captured various suspicious activities, including Nmap OS detection, SYN scans, and service enumeration.

4.3 Suricata Alerts and Detection Summary



```
Info: suricata: time elapsed 240.513s
Info: counters: Alerts: 78
Notice: device: enp0s8: packets: 3874, drops: 0 (0.00%), invalid chksum: 0
```

Figure 9 Suricata Detection Summary

- During testing, Suricata captured various suspicious activities, including Nmap OS detection, SYN scans, and service enumeration.

- The alert counters confirmed that the Intrusion Detection System successfully generated a total of 78 alerts against the target traffic, demonstrating the IDS's ability to identify probing attempts.

- These logs demonstrate that the IDS can detect attempts to probe system vulnerabilities, allowing timely intervention before an attacker can exploit any service. By monitoring both payload content and network patterns, Suricata ensures that even subtle reconnaissance attempts are logged.

4.4 Traffic Capture Verification

```
vboxuser@A2Z:~$ sudo tcpdump -i enp0s8 -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
19:03:45.988364 IP 192.168.56.16.3000 > 192.168.56.15.50458: Flags [P.], seq 490
104243:490104246, ack 128173706, win 505, options [nop,nop,TS val 1315183716 ecr
 4282846271], length 3
19:03:45.992788 IP 192.168.56.15.50458 > 192.168.56.16.3000: Flags [P.], seq 1:8
```

Figure 10 TCPDump Traffic Capture

Traffic Capture Verification tcpdump was executed on the Ubuntu server's DMZ interface to capture and inspect raw network packets. This process confirmed that all traffic, including scanning attempts and normal web requests to the Juice Shop on port 3000, was being logged at the network layer. This validates that the monitoring infrastructure is functional and capable of capturing detailed traffic for analysis, supporting both reactive incident response and proactive security measures.

4.5 Vulnerability Assessment and Recommended Improvements

Vulnerability	Impact/Risk	Recommended Improvement
Unencrypted HTTP Traffic	High - vulnerable to MITM attacks	Enforce HTTPS with Nginx reverse proxy
Direct Exposure of Juice Shop	High - intentionally vulnerable app	Restrict port 3000 to trusted IPs / use WAF

Software Version Disclosure	Medium - helps attackers find exploits	Mask Nginx version numbers
High IDS Alert Volume	Low - operational risk	Integrate Suricata logs with SIEM

The security posture of the A2Z Corporation system can be significantly advanced by focusing on three key areas. First, address the High risk of data interception by implementing a Nginx reverse proxy to enforce HTTPS across all external communication, providing encryption for data in transit. Second, mitigate the risk posed by the intentionally vulnerable application by placing a Web Application Firewall (WAF) in front of the application and restricting direct access to its native port (3000/tcp) via UFW rules. Finally, improve operational security by integrating Suricata logs with a SIEM system (Security Information and Event Management) to centralize alert correlation, filter noise, and ensure timely response to genuine threats, transforming high alert volumes into actionable security intelligence.

5.0 Conclusion

This project successfully implemented a secure infrastructure for A2Z Corporation's internet-facing systems. The layered security design, combining DMZ segmentation, UFW firewall, Docker containerization, and Suricata IDS, effectively mitigates key risks such as unauthorized access, reconnaissance, service exploitation, and lateral movement. Security measures were validated through penetration testing and network monitoring. Nmap scans confirmed that only necessary services were accessible, Suricata produced real-time alerts for all scanning activity, and tcpdump verified traffic visibility.

The system reduces the attack surface, isolates vulnerable applications, and provides continuous monitoring, aligning with best practices in cybersecurity. Recommendations such as HTTPS enforcement, WAF placement, port restrictions, and SIEM integration further enhance protection.

In conclusion, the project demonstrates a full lifecycle of system hardening: from design, implementation, testing, to monitoring. A2Z Corporation's internet-facing services are now more resilient, better monitored, and secure against external threats, providing both operational continuity and enhanced cybersecurity assurance.

6.0 Git Hub Link

Link: https://github.com/PatrickRogersSCSJ2100424/Cyber-Security-Report.git

7.0 References

Docker. (2020, August 14). *Install Docker Engine on Ubuntu*. Docker Documentation. https://docs.docker.com/engine/install/ubuntu/

Ubuntu. (2023, September 27). *UFW - Community Help Wiki*. Help.ubuntu.com. https://help.ubuntu.com/community/UFW

Boucheron, B., Camisso, J., & Abid, E. (2024, February 27). *How to Set Up a Firewall with UFW on Ubuntu | DigitalOcean*. Www.digitalocean.com. https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu

OWASP. (2025). *Juice Shop - Insecure Web Application for Training | OWASP*. Owasp.org. https://owasp.org/www-project-juice-shop/