



GROUP 8

Dec 03, 2023

PROJECT – VULNERABILITY ASSESSMENT

ISMAIL MAHAMED 125052191

JASKARAN SOHAL 150343218

RAYYAN KHAN 155534209

EASTON SOARES 108851213

Table of Contents

Executive Summary.....	4
Scope	4
Testing Details	5
Security Strategic Recommendation.....	5
Risk Assessment Overview.....	6
Vulnerabilities & Manual Testing / Validation.....	7
High Vulnerabilities	7
Fuzzing – Wfuzz.....	8
Tools Used	9

Figure 1: Number of Vulnerabilities discovered on the network.....	4
Figure 2:Testing Detail Process.....	5
Figure 3: Risk Overview Diagram	6
Figure 4: Wfuzz Network Services.....	8
Figure 5: Wfuzz Ecom Site	8
Figure 6: Wfuzz Corpsite	9
Figure 7: Nessus Scan.....	9
Figure 8: Nessus Scan.....	10
Figure9: OpenVAS scan	10
Figure 10: NMAP scan	10
Figure 11: Nmap Scans.....	11
Figure 12: Skipfish Corp	11
Figure 13: Skipfish Ecom	11
Figure 14: Zaproxy.....	12
Figure 15: Zaproxy Scan	12
Table 1: Network Scope	4
Table 2: Network Devices.....	4
Table 3: Level of Risk	6
Table 4: PoC table.....	8

Executive Summary

This report presents findings from a comprehensive vulnerability assessment conducted on a network topology consisting of 10 devices, distributed across Management, DMZ, and Internal zones. Leveraging OpenVAS, Nessus, and Skip fish, we identified a total of 15 vulnerabilities categorized as 4 highs, 9 mediums, and 2 lows.

High	Medium	Low
4	9	2

Figure 1: Number of Vulnerabilities discovered on the network.

High-severity vulnerabilities demand immediate attention to fortify critical aspects of the network, while addressing medium-severity issues is crucial for overall robustness. The report recommends prompt remediation actions, regular vulnerability scanning, and zone-specific security measures to mitigate risks effectively.

Implementation of these recommendations will significantly enhance the network's security posture, ensuring resilience against potential cyber threats. Ongoing vigilance and proactive security measures are essential for maintaining a secure and reliable network environment.

Scope

Networks
192.168.9.192/26
192.168.9.64/26
192.168.9.128/26

Table 1: Network Scope

Network Devices
Core-Router
External Router
User
Docker Containers
Active Directory
Network Services & Public DNS
Private DNS

Table 2: Network Devices

Testing Details

The journey from the initial initiation of the process to its transformation into a TVRA document after scanning and exploitation.



Figure 2: Testing Detail Process

Security Strategic Recommendation

key Areas for Improvements

1. **Strengthening Access Control and Password Management:** Implementing robust measures to safeguard against SSH brute force login attempts with default credentials is vital. It is advisable to enforce stringent password policies, promote regular password changes, and explore the adoption of multi-factor authentication (MFA) for heightened access security. Additionally, it is recommended that the client conducts a thorough review to disable default credentials across systems and services, thereby mitigating the risk of unauthorized access.
2. **Service Hardening for Enhanced Security:** Taking steps to bolster security involves disabling or restricting unnecessary services, such as the rlogin service, which can pose potential risks. Only essential services should be enabled and actively maintained. Regular assessments of running services on systems are crucial to ensure alignment with security best practices. If a service is deemed unnecessary, it should be promptly disabled to minimize the overall attack surface.
3. **Network Segmentation and Access Control Optimization:** Consideration should be given to implementing effective network segmentation to isolate sensitive systems and services from less secure segments of the network. The strategic use of firewalls and access controls becomes imperative, particularly in restricting access to critical services like SMB. Configuring SMB accounts with robust authentication measures is essential, and any unnecessary SMB services should be disabled. Regular audits and reviews of access permissions are vital for minimizing potential vulnerabilities.

Risk Assessment Overview

A risk assessment was conducted by evaluating both the likelihood and potential impact of vulnerabilities. This assessment allowed for the determination of the overall risk associated with these vulnerabilities. The calculations were performed using the NIST 800-31 table as a reference.

TABLE I-2: ASSESSMENT SCALE – LEVEL OF RISK (COMBINATION OF LIKELIHOOD AND IMPACT)

Likelihood (Threat Event Occurs and Results in Adverse Impact)	Level of Impact				
	Very Low	Low	Moderate	High	Very High
Very High	Very Low	Low	Moderate	High	Very High
High	Very Low	Low	Moderate	High	Very High
Moderate	Very Low	Low	Moderate	Moderate	High
Low	Very Low	Low	Low	Low	Moderate
Very Low	Very Low	Very Low	Very Low	Low	Low

Table 3: Level of Risk

After concluding the risk assessment and evaluating the risk levels linked to each vulnerability, a diagram was created to depict the ranking of these vulnerabilities. The prioritization of vulnerabilities was predominantly influenced by their severity, as indicated by the security tool's assessments.

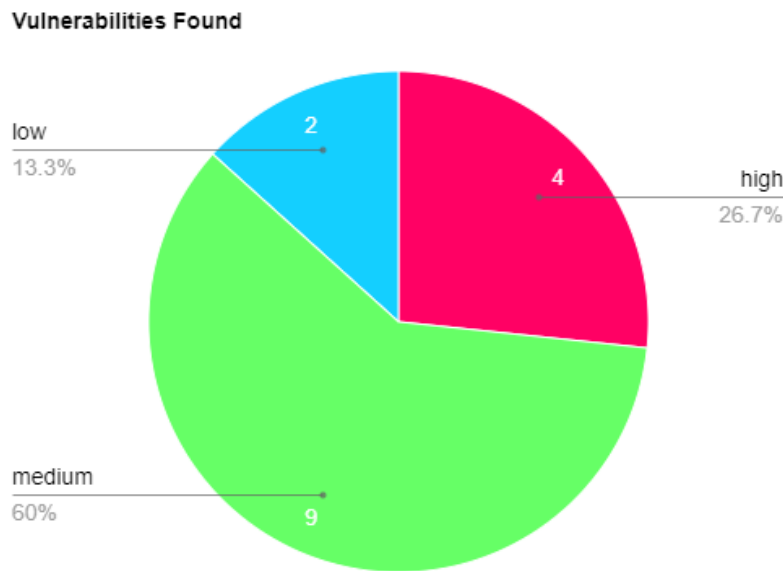
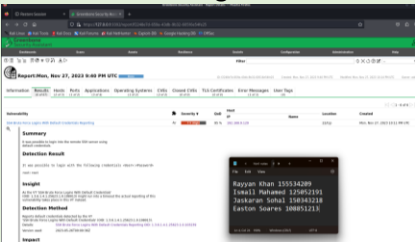
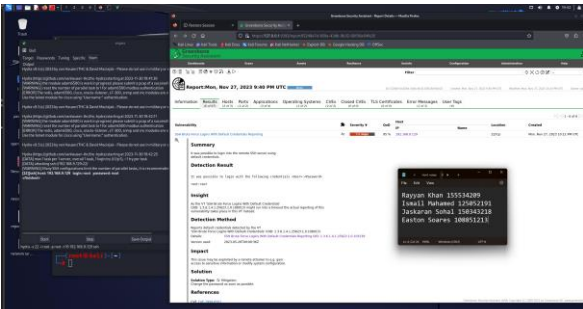
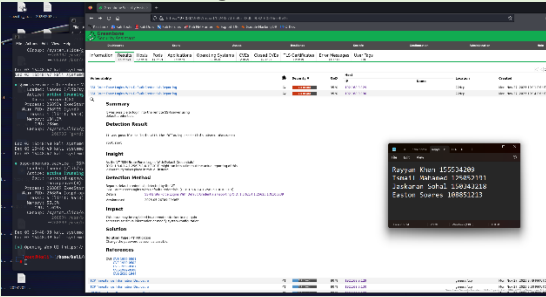


Figure 3: Risk Overview Diagram

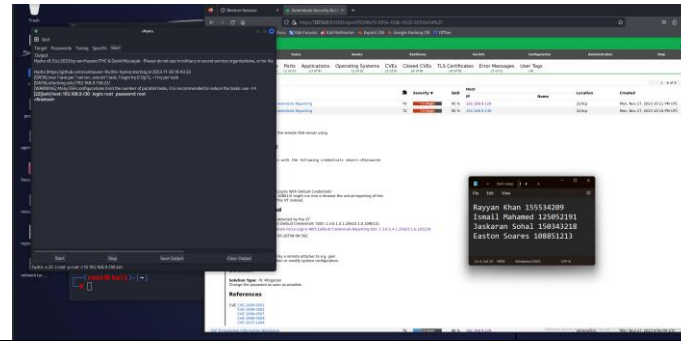
Vulnerabilities & Manual Testing / Validation

High Vulnerabilities

Scope Affected	192.168.9.129
Vulnerability	SSH brute force logins with default credentials Reporting 
Level of Impact	High
Overall Likelihood	High
Exploitation	

Scope Affected	192.168.9.130
Vulnerability	SSH Brute force logins with default credentials reporting 
Level of Impact	High
Overall Likelihood	High

Exploitation



Fuzzing – Wfuzz

As part of our testing methodology, we leveraged Wfuzz, a fuzz testing tool, to evaluate the robustness of the network services and applications. Wfuzz will allow us to discover potential vulnerabilities by fuzzing parameters and payloads.

The following is a brief proof of concept where fuzzing was applied to test vulnerabilities:

Scope	192.168.9.194, 172.17.0.2, 172.17.0.3
Vulnerability	HTTP TRACE
Level of Impact	Low

Table 4: PoC table

Wfuzz on Network Services

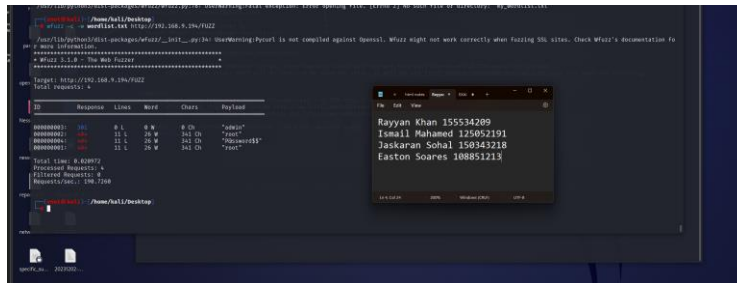


Figure 4: Wfuzz Network Services

Wfuzz Ecommerce site

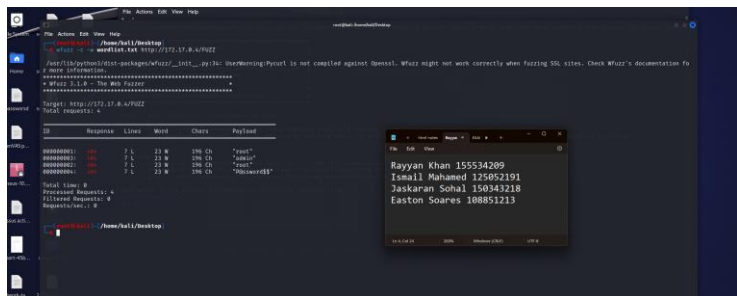


Figure 5: Wfuzz Ecom Site

Wfuzz on corpsite

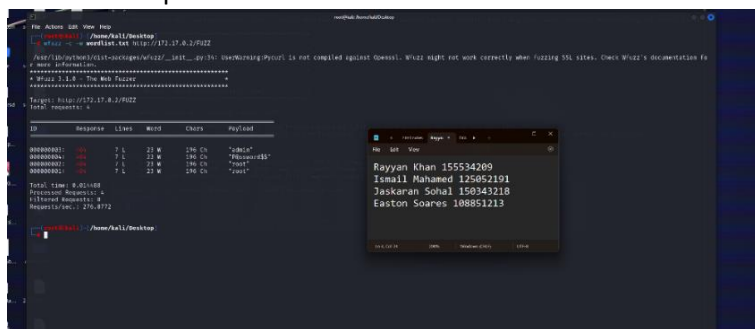


Figure 6: Wfuzz Corpsite

Our fuzzing tests targeting the Network Services machine have revealed a security vulnerability related to the active HTTP TRACE method, which poses a potential risk for Cross-Site Tracing (XST) attacks. Exploiting this functionality could lead to unauthorized access to sensitive data.

This security concern extends to our corporate site and e-commerce platform as well. To address this issue promptly, we recommend a comprehensive mitigation strategy.

The first step involves disabling the HTTP TRACE method on all web servers, including those supporting our corporate and e-commerce sites. This requires updating server configurations and ensuring that this method is deactivated in future settings.

Tools Used

In conducting the vulnerability assessment, three tools—OpenVAS, Nessus, and Skip fish—were used to comprehensively analyze the security landscape of the network topology. The scans were initiated from a Kali machine designated as the attacker machine, simulating an external perspective. The results revealed a total of 15 vulnerabilities, categorized as 4 high-severity, 9 medium-severity, and 2 low-severity issues across the network. This approach not only identified specific vulnerabilities but also provided insights into potential risks, enabling targeted and informed remediation efforts.

The provided screenshots below serve as concrete evidence of the vulnerability scans taking place.

Nessus Scans

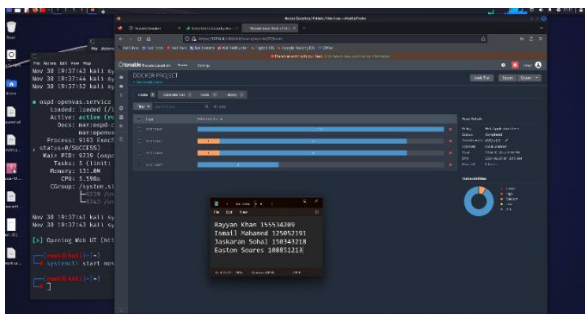


Figure 7: Nessus Scan

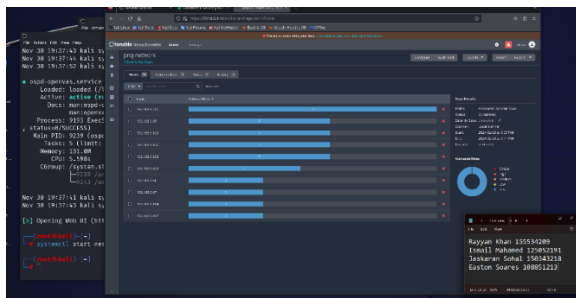


Figure 8: Nessus Scan

OpenVAS Scans

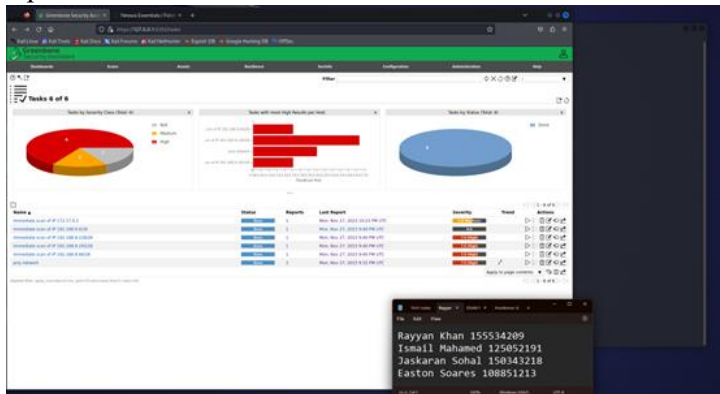


Figure9: OpenVAS scan

Nmap Scan

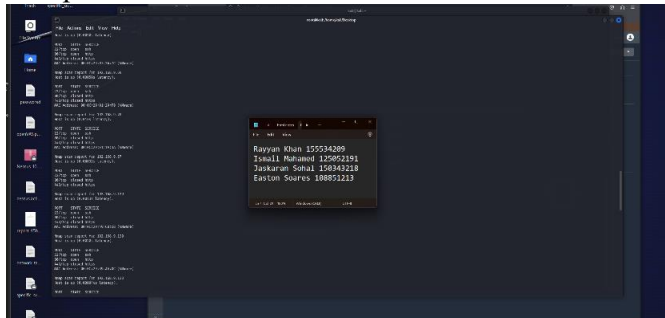


Figure 10: NMAP scan

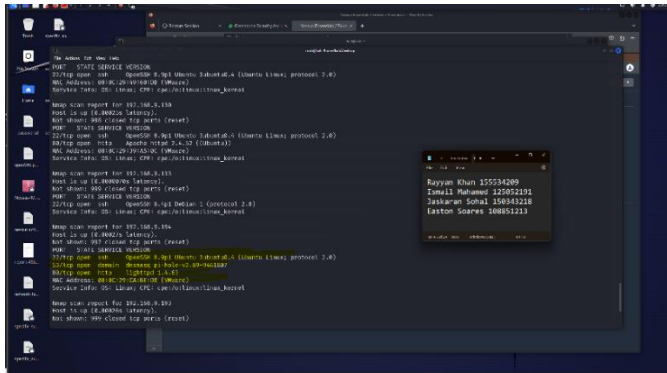


Figure 11: Nmap Scans

Skipfish

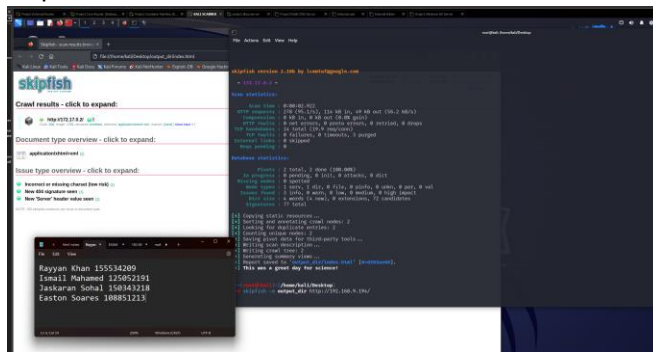


Figure 12: Skipfish Corp

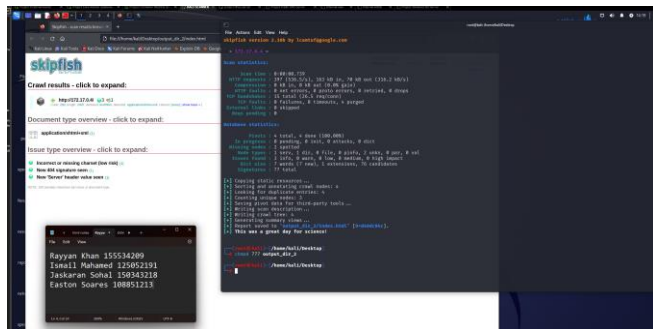


Figure 13: Skipfish Ecom

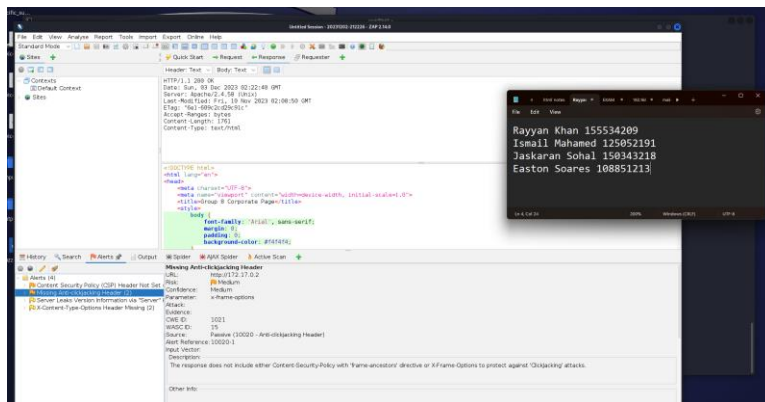


Figure 14: Zaproxy

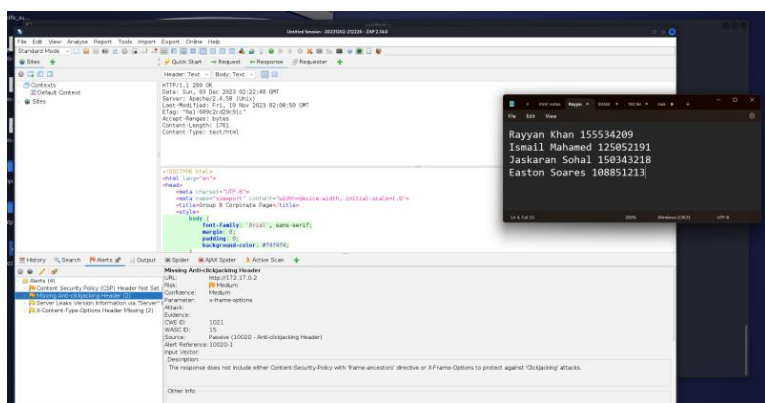


Figure 15: Zaproxy Scan