RED TEAM SECURITY ASSESSMENT REPORT

Title: Red Team Security Assessment Report
Target System: Metasploitable2 (Lab Environment)
Attacker System: Kali Linux
Prepared By: *Utkarsh Kumar*
Purpose: Academic and Training Use Only

1. Executive Summary

A Red Team security assessment was conducted on a vulnerable lab-based system to evaluate its security posture by simulating real-world attacker behavior. The assessment identified multiple exposed services and critical vulnerabilities that could allow unauthorized access to the system. Through reconnaissance, vulnerability scanning, and exploitation, successful system compromise was achieved, demonstrating the risks associated with outdated services and weak configurations. The findings emphasize the importance of regular security assessments, timely patch management, strong authentication mechanisms, and continuous monitoring to reduce the attack surface and mitigate potential threats.

2. Objective of the Assessment

The objective of this Red Team exercise was to simulate an attacker's approach in a controlled environment by identifying open services, discovering vulnerabilities, exploiting identified weaknesses, and performing post-exploitation activities. This assessment aimed to enhance practical understanding of Red Team methodologies, tools, and reporting practices while highlighting the security risks present in poorly secured systems.

3. Scope and Rules of Engagement (RoE)

Scope

- **In-Scope Systems**
    - Metasploitable2 virtual machine
    - Windows test virtual machine (for password and persistence tasks)
- **Out-of-Scope Systems**
    - Host operating system

- Internet-facing or real production systems

Rules of Engagement

- Allowed activities included reconnaissance, network scanning, vulnerability assessment, exploitation, and post-exploitation.
- Destructive actions such as data deletion, denial-of-service attacks, or modification of critical system files were strictly prohibited.
- The assessment was conducted solely for educational purposes in an isolated lab environment.

4. Environment Setup

The assessment was performed in a controlled virtual lab environment to ensure ethical and safe testing. Kali Linux was used as the attacker machine, while Metasploitable2 served as the vulnerable target system. Both virtual machines were configured within the same isolated network to allow controlled communication while preventing access to external systems. Network connectivity between the machines was verified before beginning the assessment.

5. Reconnaissance and Network Scanning

5.1 Purpose of Reconnaissance

Reconnaissance is a critical initial phase of a Red Team assessment, aimed at identifying the attack surface of the target system. The objective of this phase is to gather information about open ports, running services, and service versions without directly exploiting the system. Effective reconnaissance helps attackers understand potential entry points and select appropriate vulnerabilities for further exploitation while minimizing detection.

5.2 Tool Used

- **Nmap (Network Mapper)**

Nmap was selected for this assessment due to its reliability and ability to perform various types of scans, including stealth scans, service version detection, script-based enumeration, and aggressive scans.

5.3 Types of Scans Performed

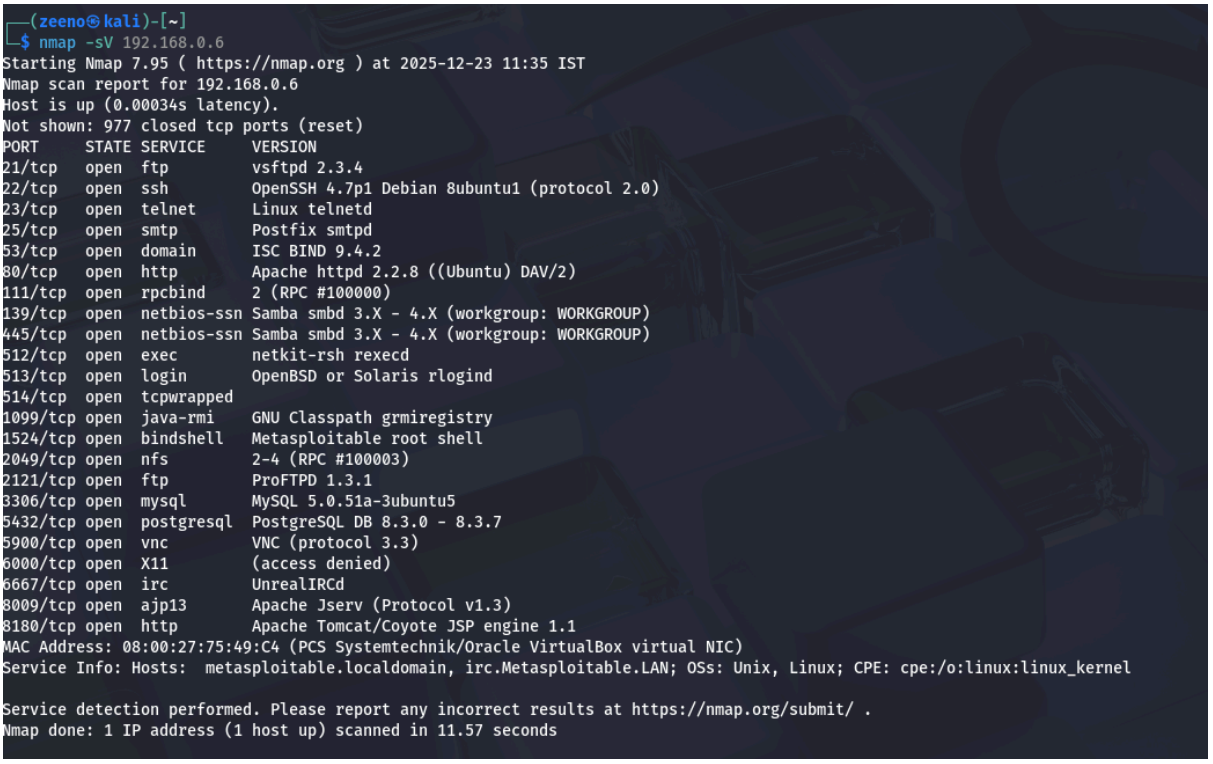
The following scans were performed to analyze the target system in detail:

Service and Version Detection Scan

Identifies open ports along with the services and their running versions.

Command Used:

nmap -sV 192.168.1.6



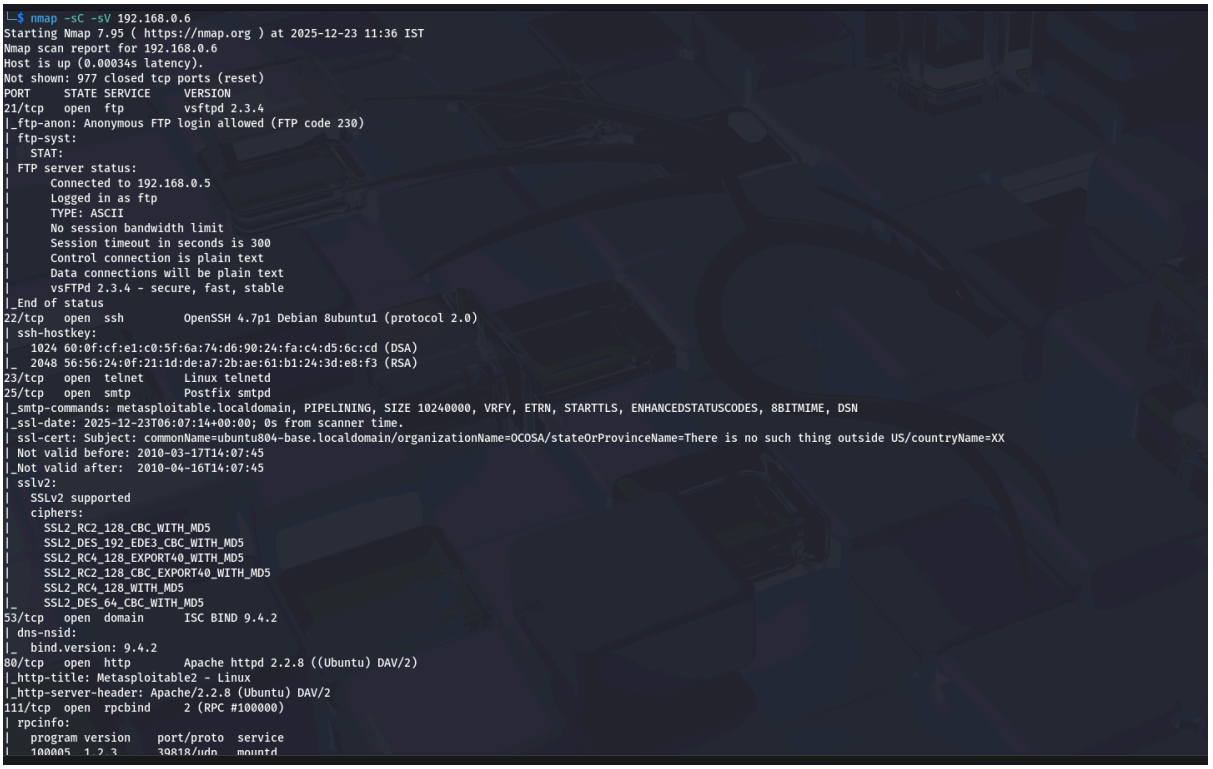
Service & Version Detection Scan Output

Default Script and Service Enumeration Scan

Executes default Nmap scripts to gather additional information such as banners and configuration details.

Command Used:

nmap -sC -sV 192.168.1.6



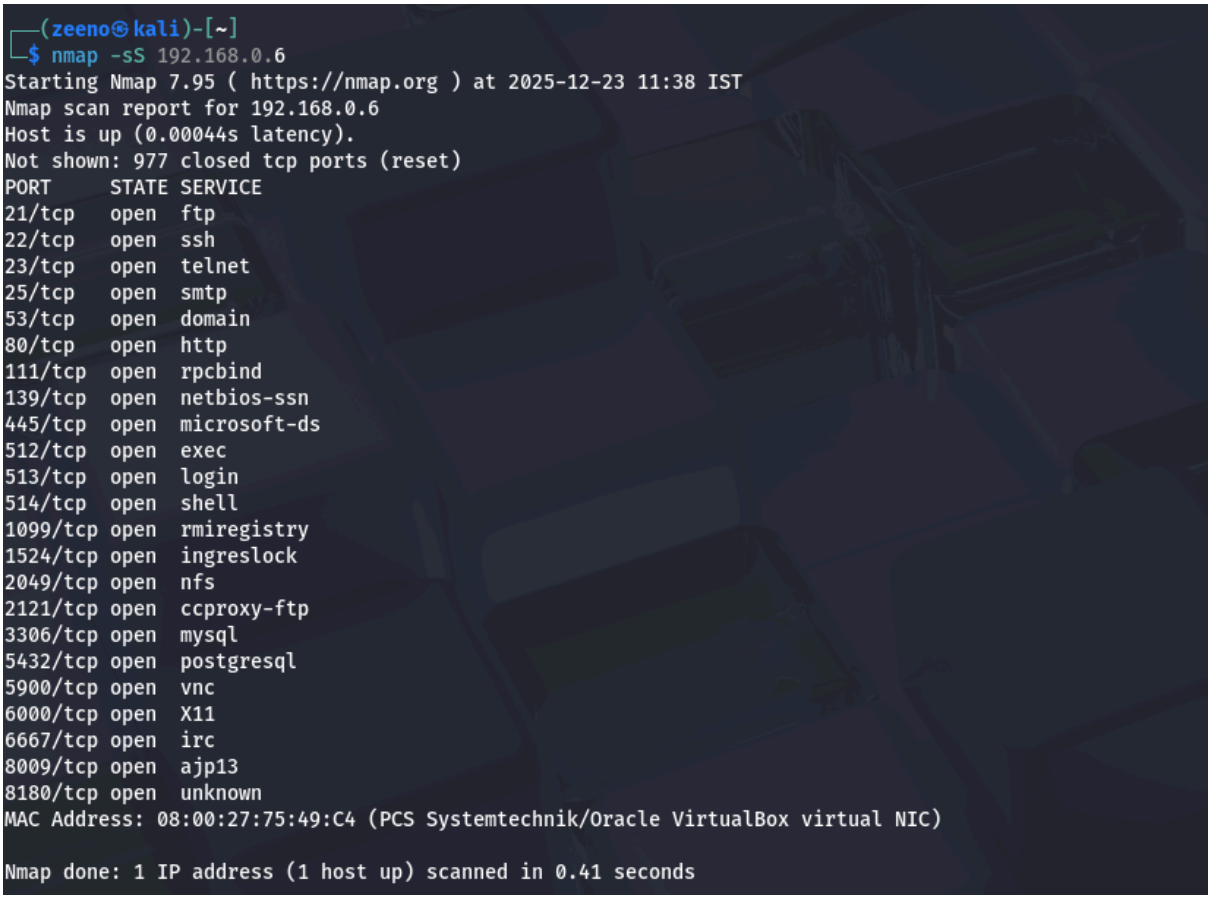
Script & Service Enumeration Scan Output

Stealth (SYN) Scan

Performs a half-open scan to reduce detection by security monitoring systems.

Command Used:

nmap -sS 192.168.1.6



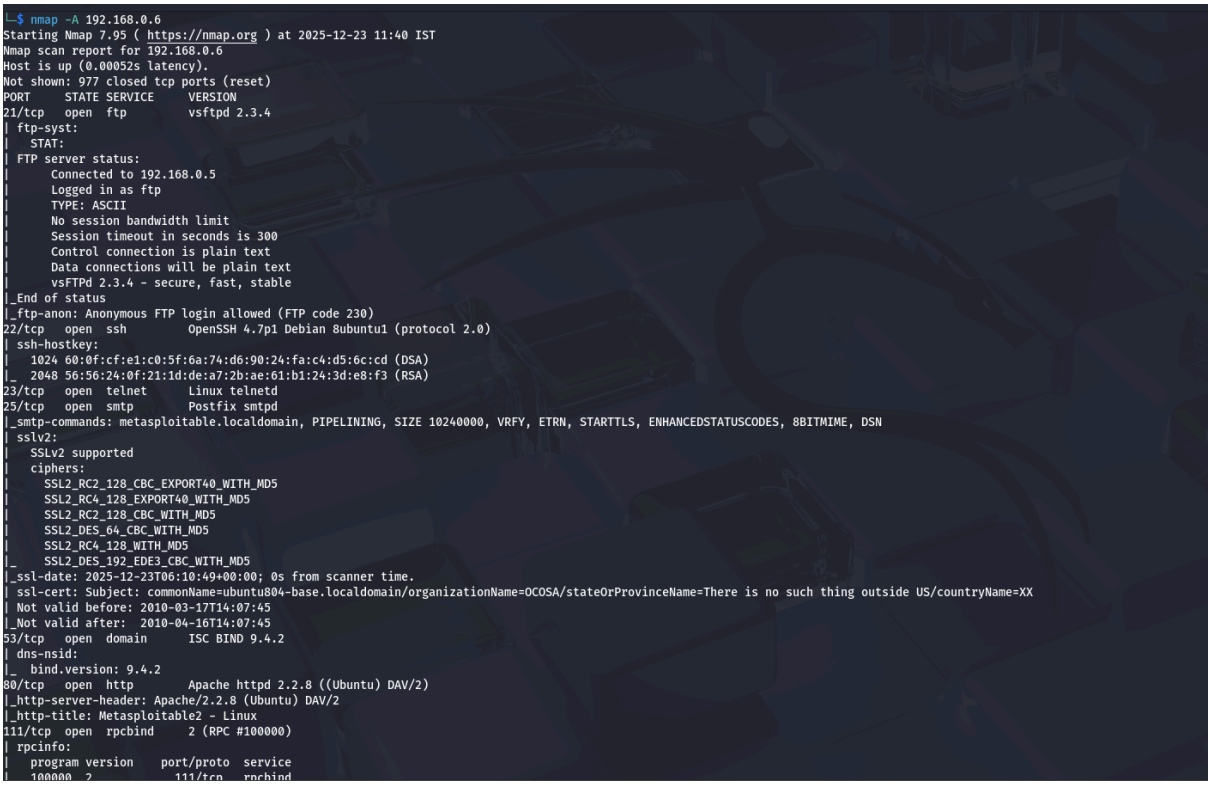
Stealth Scan Output

Aggressive Scan

Combines OS detection, version detection, script scanning, and traceroute to provide extensive target information.

Command Used:

nmap -A 192.168.1.6



(Aggressive Scan Output)

5.4 Scan Results Summary

The reconnaissance scans revealed multiple open ports and running services that increase the system's attack surface. The following table summarizes key findings:

Findings

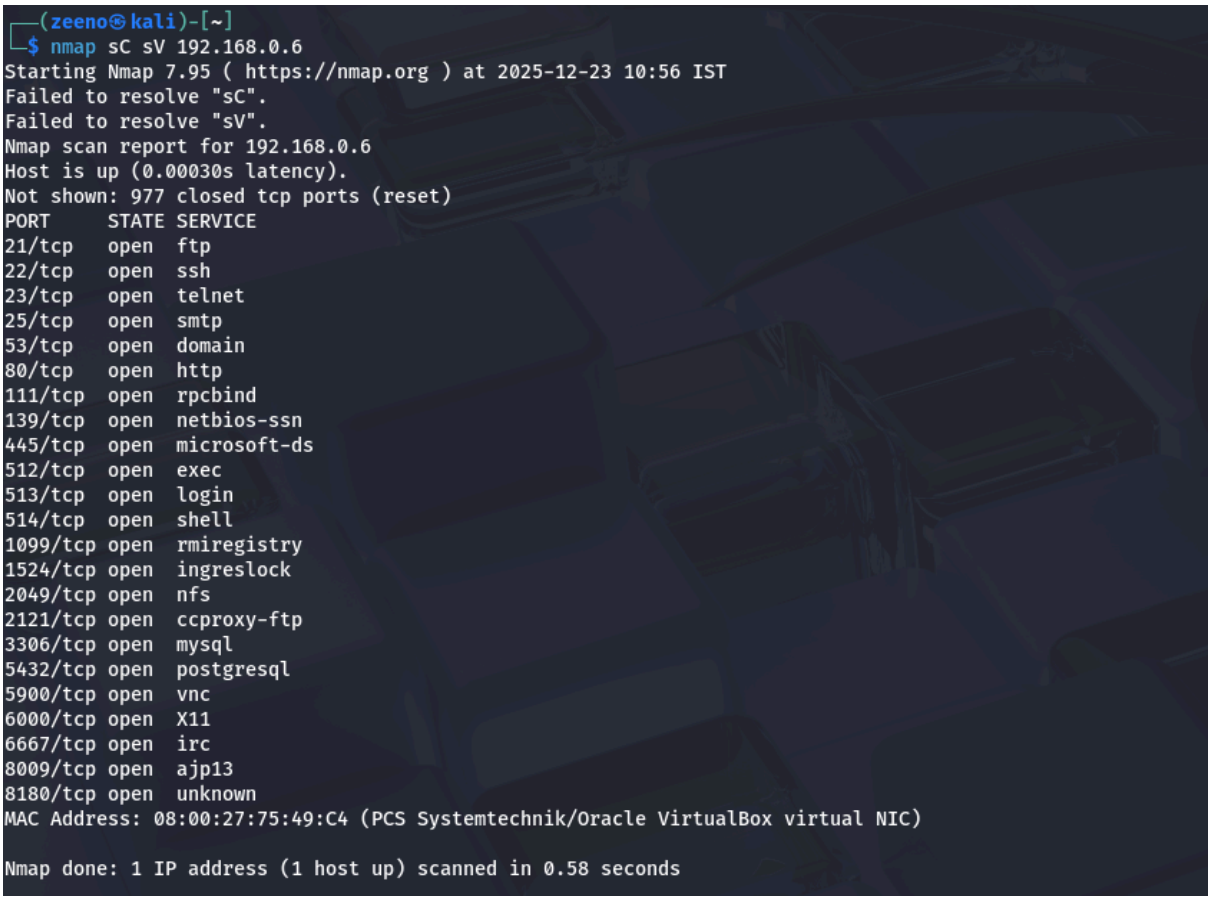
PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
25/tcp	open	smtp

53/tcp	open	domain
80/tcp	open	http
111/tcp	open	rpcbind
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
1099/tcp	open	rmiregistry
1524/tcp	open	ingreslock
2049/tcp	open	nfs
2121/tcp	open	ccproxy-ftp
3306/tcp	open	mysql

5432/tcp	open	postgresql
5900/tcp	open	vnc
6000/tcp	open	X11
6667/tcp	open	irc
8009/tcp	open	ajp13
8180/tcp	open	unknown

Scan Analysis

A stealth scan (-sS) was observed to be less intrusive and slower, while an aggressive scan (-A) provided more detailed information including service versions and scripts but was more detectable. Aggressive scans increase visibility and are more likely to trigger security alerts.



Nmap scan output

6. Vulnerability Assessment

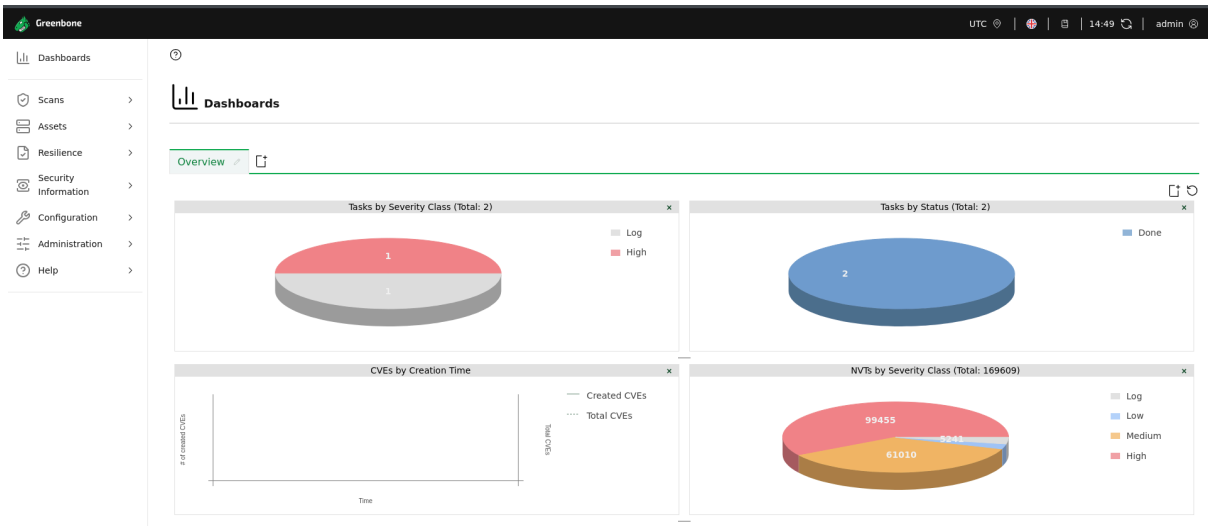
6.1 Purpose of Vulnerability Assessment

Vulnerability assessment is a critical phase in a Red Team engagement that focuses on identifying known security weaknesses in a target system without actively exploiting them. The purpose of this phase is to evaluate the system's exposure to publicly known vulnerabilities and to prioritize risks based on severity and potential impact. This information is later used to select suitable vulnerabilities for controlled exploitation.

6.2 Tool Used

- **OpenVAS (Greenbone Vulnerability Manager)**

OpenVAS was selected for this assessment due to its extensive vulnerability database, automated scanning capabilities, and CVSS-based risk scoring, which allows vulnerabilities to be ranked according to their severity.



OpenVAS Dashboard after login

6.3 Scan Configuration and Execution

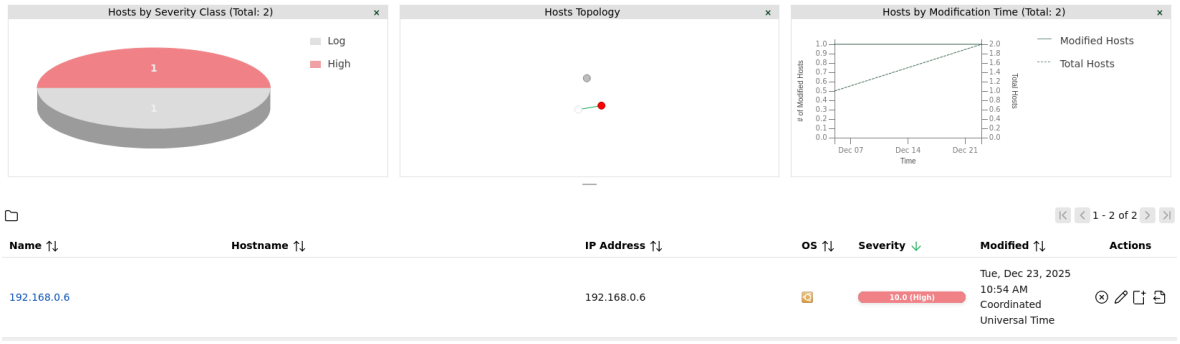
The vulnerability scan was configured to perform a comprehensive analysis of the target system.

Scan Configuration Details:

- **Scan Target:** Metasploitable2
- **Scanner:** OpenVAS Default

- **Scan Profile:** Full and Fast
- **Scanning Method:** Network-based vulnerability scanning

Once configured, the scan was executed and monitored until completion.



Scan completed successfully

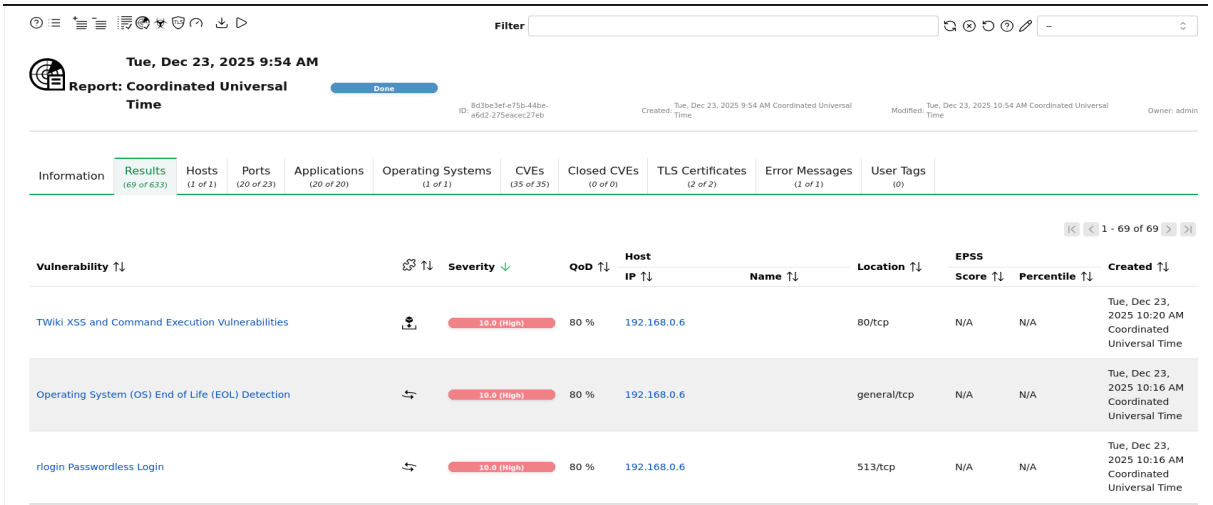
6.4 Critical and High-Risk Vulnerabilities Identified

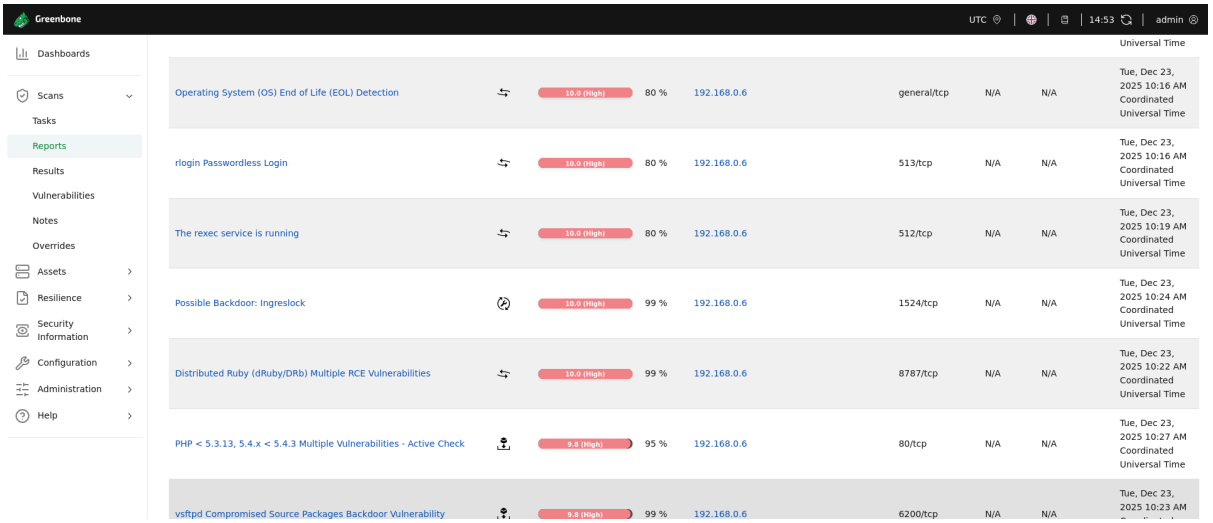
The OpenVAS vulnerability scan identified a large number of critical and high-severity vulnerabilities on the target system (192.168.0.6). Many of these vulnerabilities are associated with outdated services, default credentials, insecure configurations, and known backdoors. The presence of multiple vulnerabilities with CVSS scores close to 10 indicates that the system is highly exposed and can be compromised with minimal effort.

Key High-Risk Vulnerabilities (Prioritized)

Vulnerability	Affected Port	CVSS Score	Risk Level	Description
Operating System End of Life (EOL) Detection	General	10.0	Critical	The operating system is no longer supported, making it vulnerable to unpatched exploits
rlogin Passwordless Login	513/tcp	10.0	Critical	Allows login without authentication
rexec Service Running	512/tcp	10.0	Critical	Enables remote command execution

				without encryption
Possible Backdoor: Ingreslock	1524/tcp	10.0	Critical	Indicates a potential backdoor allowing unauthorized access
Distributed Ruby (dRuby) RCE	8787/tcp	10.0	Critical	Allows remote code execution
vsftpd Compromised Backdoor	21/tcp, 6200/tcp	9.8	High	Provides unauthorized shell access
Apache Tomcat AJP RCE (Ghostcat)	8009/tcp	9.8	High	Allows remote file inclusion and command execution
MySQL Default Credentials	3306/tcp	9.8	High	Database accessible using default login credentials
DistCC RCE Vulnerability	3632/tcp	9.3	High	Allows arbitrary command execution
UnrealIRCd Authentication Spoofing	6697/tcp	8.1	High	Allows attacker to bypass authentication





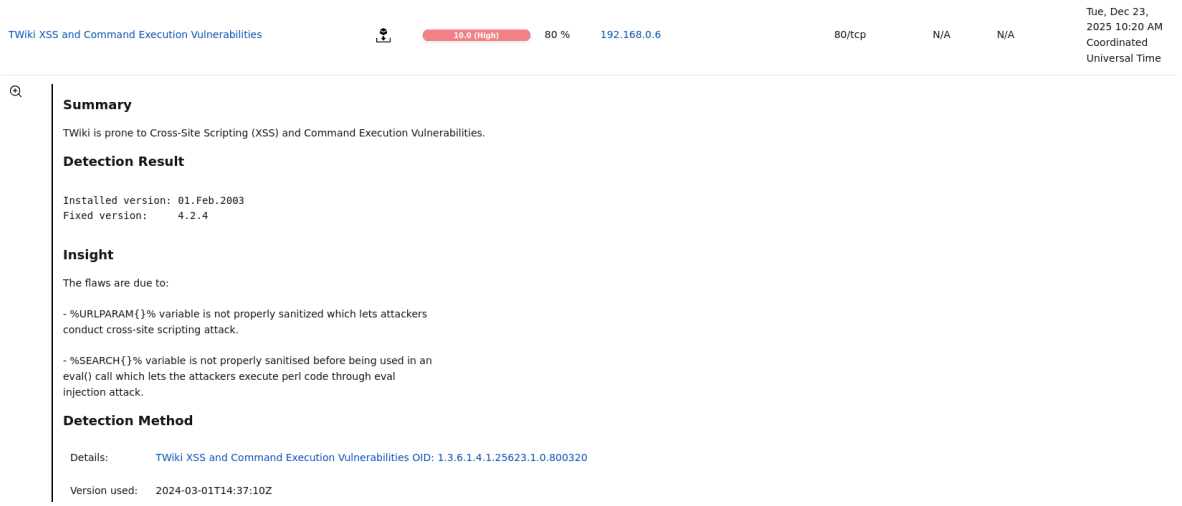
Vulnerability list

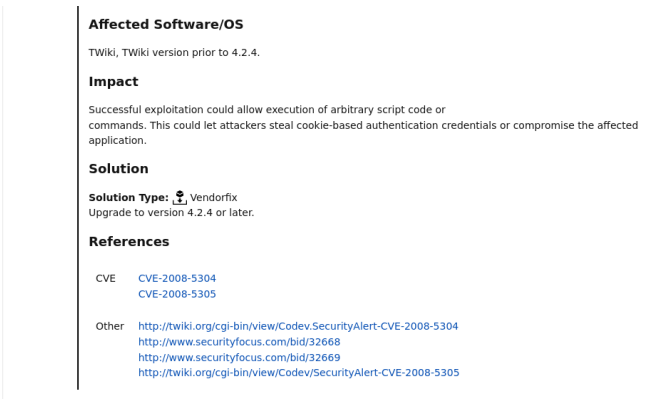
6.5 CVSS and Risk Analysis

The Common Vulnerability Scoring System (CVSS) was used to assess the severity of each identified vulnerability. Vulnerabilities with high CVSS scores indicate a greater likelihood of exploitation and a higher potential impact on system confidentiality, integrity, and availability. The presence of multiple high-risk vulnerabilities suggests that the target system is highly susceptible to compromise.

6.6 Impact Assessment

If exploited, the identified vulnerabilities could allow attackers to gain unauthorized access, execute arbitrary commands, and escalate privileges within the system. Such access may result in complete system compromise, data leakage, and persistent attacker presence, emphasizing the need for immediate remediation.





Affected Software/OS

TWiki, TWiki version prior to 4.2.4.

Impact

Successful exploitation could allow execution of arbitrary script code or commands. This could let attackers steal cookie-based authentication credentials or compromise the affected application.

Solution

Solution Type: Vendorfix
Upgrade to version 4.2.4 or later.

References

CVE	CVE-2008-5304 CVE-2008-5305
Other	http://twiki.org/cgi-bin/view/Codev.SecurityAlert-CVE-2008-5304 http://www.securityfocus.com/bid/32668 http://www.securityfocus.com/bid/32669 http://twiki.org/cgi-bin/view/Codev/SecurityAlert-CVE-2008-5305

Detailed vulnerability impact view

7. Exploitation

7.1 Purpose of Exploitation

The exploitation phase aims to validate whether the vulnerabilities identified during the vulnerability assessment can be practically abused by an attacker. Unlike vulnerability scanning, which only identifies potential weaknesses, exploitation demonstrates the **real-world impact** by gaining unauthorized access to the target system. This phase helps assess the severity of vulnerabilities and the level of risk they pose to system security.

7.2 Vulnerability Selected for Exploitation

Based on the vulnerability assessment results, the **VSFTPD 2.3.4 Compromised Source Packages Backdoor Vulnerability** was selected for exploitation.

Reason for Selection:

- Classified as **High/Critical severity** with a CVSS score of **9.8**
- Publicly known backdoor vulnerability
- Exploitable remotely without authentication
- Reliable exploit available in Metasploit

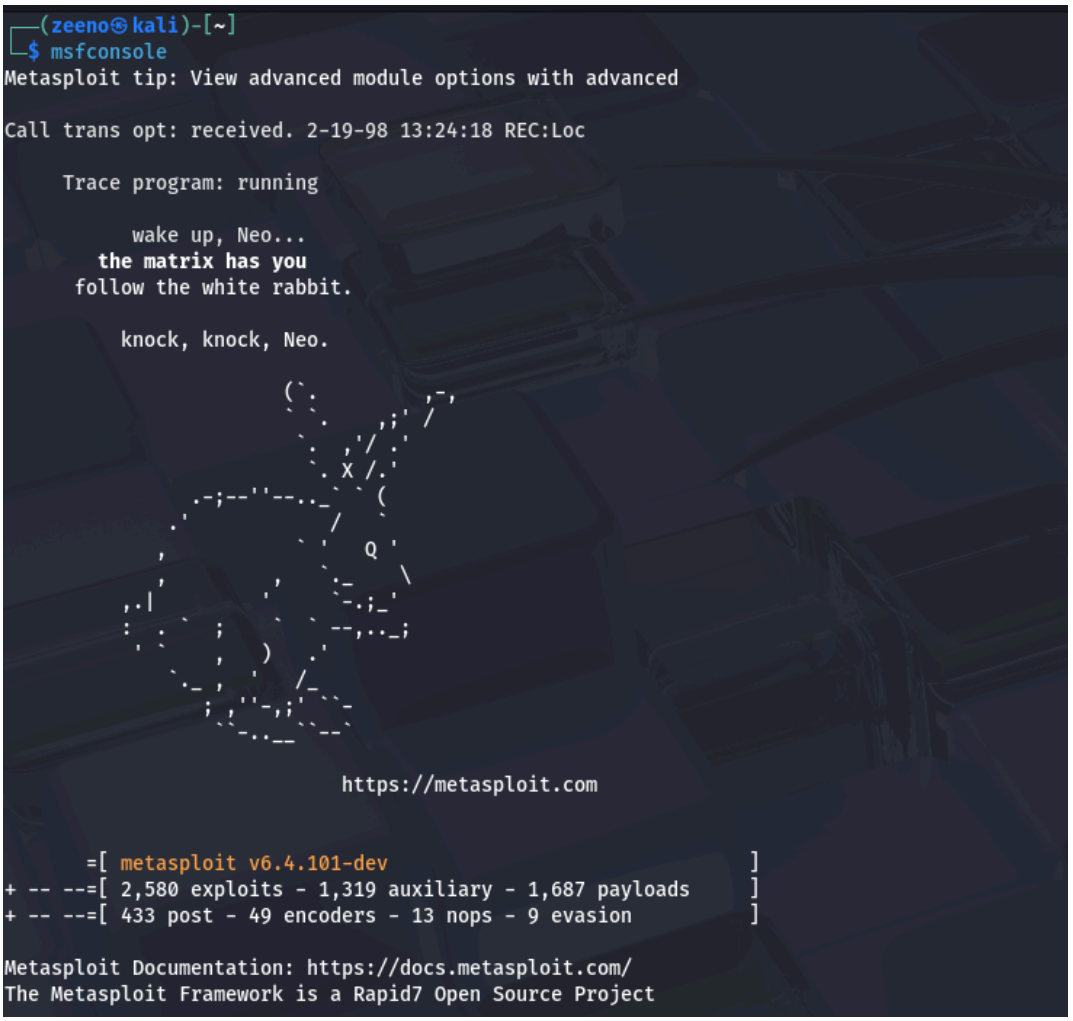
Affected Service Details:

- **Service:** FTP
- **Port:** 21/tcp
- **Target IP:** 192.168.0.6

7.3 Tool Used

- **Metasploit Framework**

Metasploit was used for exploitation due to its extensive exploit database, automation capabilities, and reliability in controlled penetration testing environments.



Metasploit Framework launched – msfconsole

7.4 Exploitation Procedure

The following steps were performed to exploit the selected vulnerability:

1. The Metasploit Framework was launched from the Kali Linux attacker machine.
2. The VSFTPD backdoor exploit module was selected.
3. The target system's IP address was configured.

4. The exploit was executed to trigger the backdoor.

5. A command shell session was successfully established.

Exploit Module Used:

- exploit/unix/ftp/vsftpd_234_backdoor

7.5 Exploitation Result and Verification

The exploitation was successful, resulting in the opening of a command shell session on the target system. Post-exploitation verification commands were executed to confirm the level of access obtained.

Verification Performed:

- whoami command confirmed **root-level access**

- uname -a confirmed access to the Metasploitable2 system

This confirms that the attacker gained **full administrative control** over the target system without authentication.

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.0.6:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.6:21 - USER: 331 Please specify the password.
[+] 192.168.0.6:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.6:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.0.5:38581 -> 192.168.0.6:6200) at 2025-12-23 19:06:31 +0530

whoami
root
whoami
uname -a
root
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Command shell session showing root access

7.6 Impact Analysis

Successful exploitation of the VSFTPD backdoor vulnerability demonstrates a **critical security failure**. An attacker exploiting this vulnerability could:

- Gain complete control over the system

- Execute arbitrary commands with root privileges

- Install malware or backdoors
- Modify or delete sensitive data
- Maintain persistent access for future attacks

The presence of such a vulnerability highlights the severe risks associated with running outdated or compromised services on production systems.

8. Post-Exploitation and Persistence

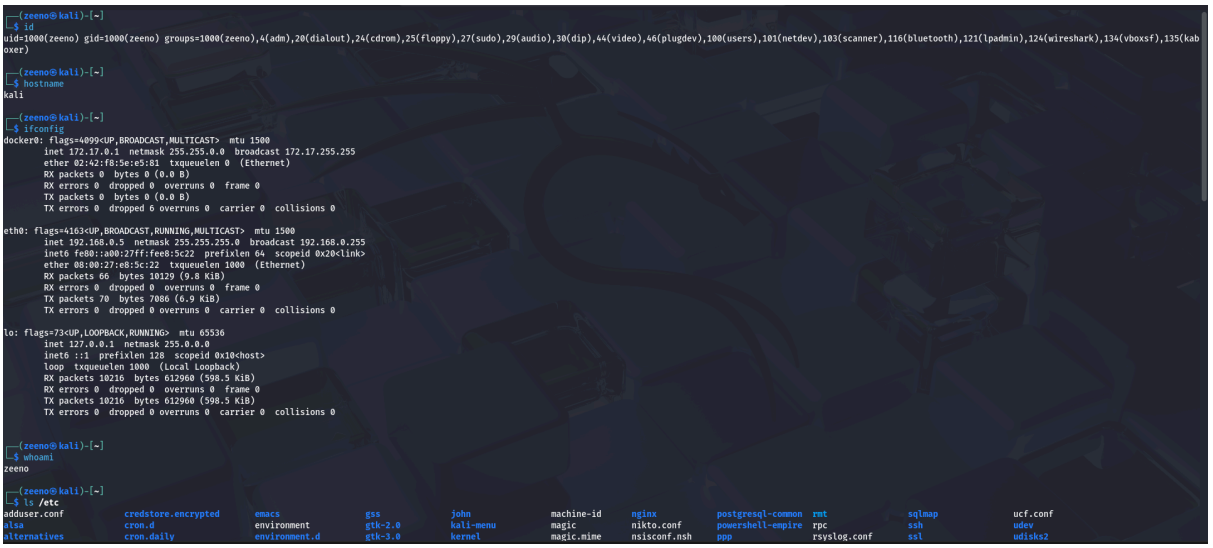
8.1 Purpose of Post-Exploitation

Post-exploitation is a critical phase of a Red Team assessment that focuses on evaluating what actions an attacker can perform after successfully compromising a system. While exploitation demonstrates initial access, post-exploitation highlights the **true impact** by showing the extent of control, access to sensitive information, and the attacker's ability to maintain long-term access. This phase helps organizations understand the consequences of a breach beyond initial compromise.

8.2 Privilege and Access Verification

After successful exploitation, the obtained shell session was used to verify the level of access gained on the target system. The attacker identity and privilege level were confirmed using standard system commands.

The results confirmed that the attacker obtained **root-level privileges**, indicating full administrative control over the system.

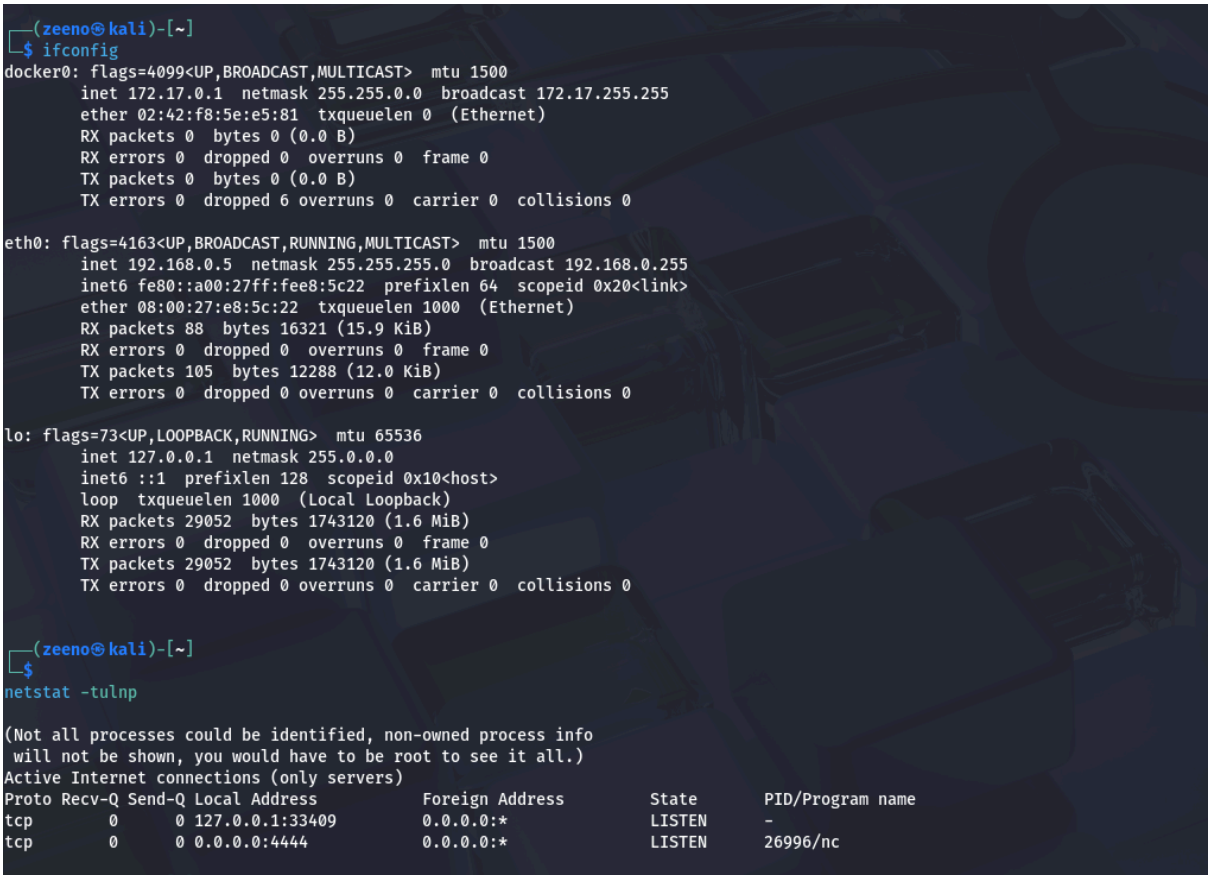


Root privilege confirmation using whoami / id commands

8.3 System and Network Enumeration

Once privileged access was confirmed, system and network enumeration was performed to gather information about the compromised host. This included identifying the operating system, hostname, and active network interfaces and services.

These actions simulate how attackers collect environmental information to plan lateral movement or further attacks within a network.

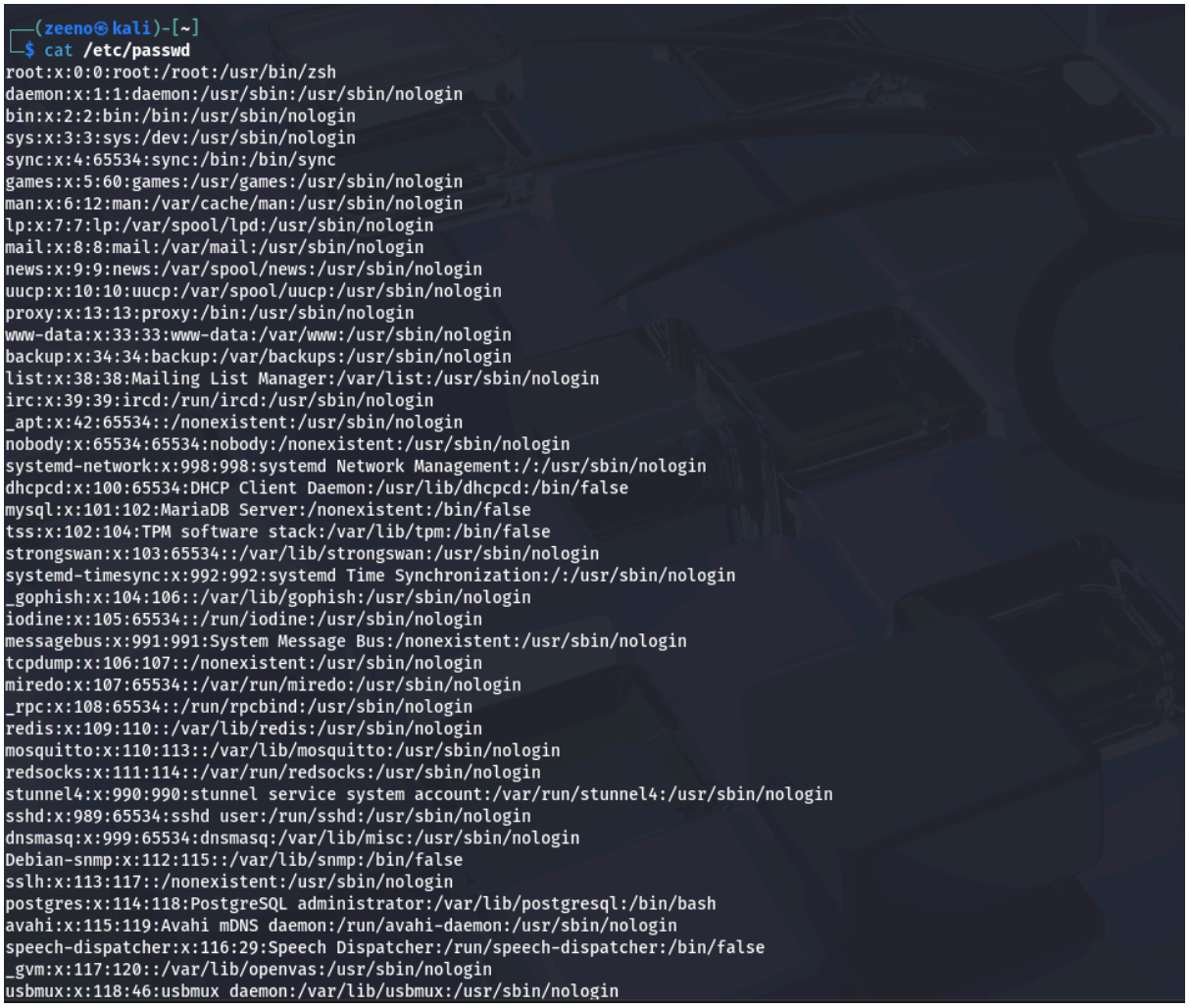


System information and network configuration output

8.4 Access to Sensitive System Information

As part of post-exploitation activities, access to sensitive system files was demonstrated in a non-destructive manner. The attacker was able to read system configuration and user account information without restriction.

This confirms a **confidentiality breach**, as attackers with root access can view critical system files and potentially harvest credentials or sensitive data.



Access to sensitive system files such as /etc/passwd

8.5 Persistence Simulation

Persistence refers to an attacker's ability to maintain access to a compromised system over time. To safely demonstrate this capability, a **non-malicious persistence simulation** was performed by creating a harmless test file on the system.

This simulation demonstrates that an attacker can write files to the system and potentially deploy persistent mechanisms such as startup scripts or scheduled tasks in real-world scenarios.

Creation and verification of persistence test file

8.6 Security Impact Analysis

The post-exploitation phase demonstrates the severe security implications of the identified vulnerabilities. With root-level access, an attacker could:

- Modify or delete critical system files
- Install malware or backdoors
- Exfiltrate sensitive data
- Maintain persistent access
- Use the compromised system to launch further attacks

9. Malware Analysis

9.1 Purpose of Malware Analysis

Malware analysis is performed to understand how security solutions detect and respond to malicious files. In a Red Team assessment, this phase demonstrates awareness of detection mechanisms used by defensive systems and helps evaluate how quickly threats can be identified. The objective of this task was to safely analyze a test file using industry-standard malware analysis platforms without executing real malware.

9.2 EICAR Test File Overview

To safely simulate malware detection, the **EICAR (European Institute for Computer Antivirus Research) test file** was used. The EICAR file contains a standardized test string that is universally recognized by antivirus engines as malicious. It does not perform any harmful action and is commonly used for training, testing, and validation of malware detection systems.

Terminal showing creation of the EICAR test file and its contents

9.3 VirusTotal Analysis

Tool Used

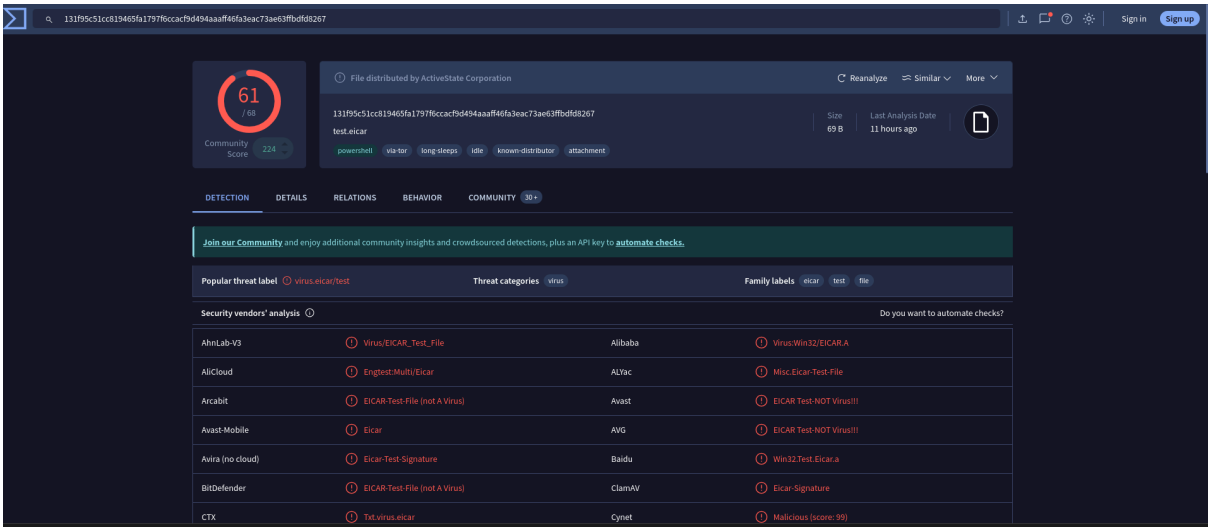
- **VirusTotal**

VirusTotal was used to perform a multi-engine antivirus scan of the EICAR test file. The file was uploaded to the VirusTotal platform, where it was scanned by numerous antivirus engines simultaneously.

Observations

- The file was detected by a majority of antivirus engines.
- The detection label identified the file as an **EICAR test file**.
- No real malicious payload or harmful activity was observed.

These results confirm that signature-based detection mechanisms are functioning correctly and can reliably identify known malicious patterns.



VirusTotal detection summary showing detection ratio

9.4 Sandbox Analysis Using Hybrid Analysis

Tool Used

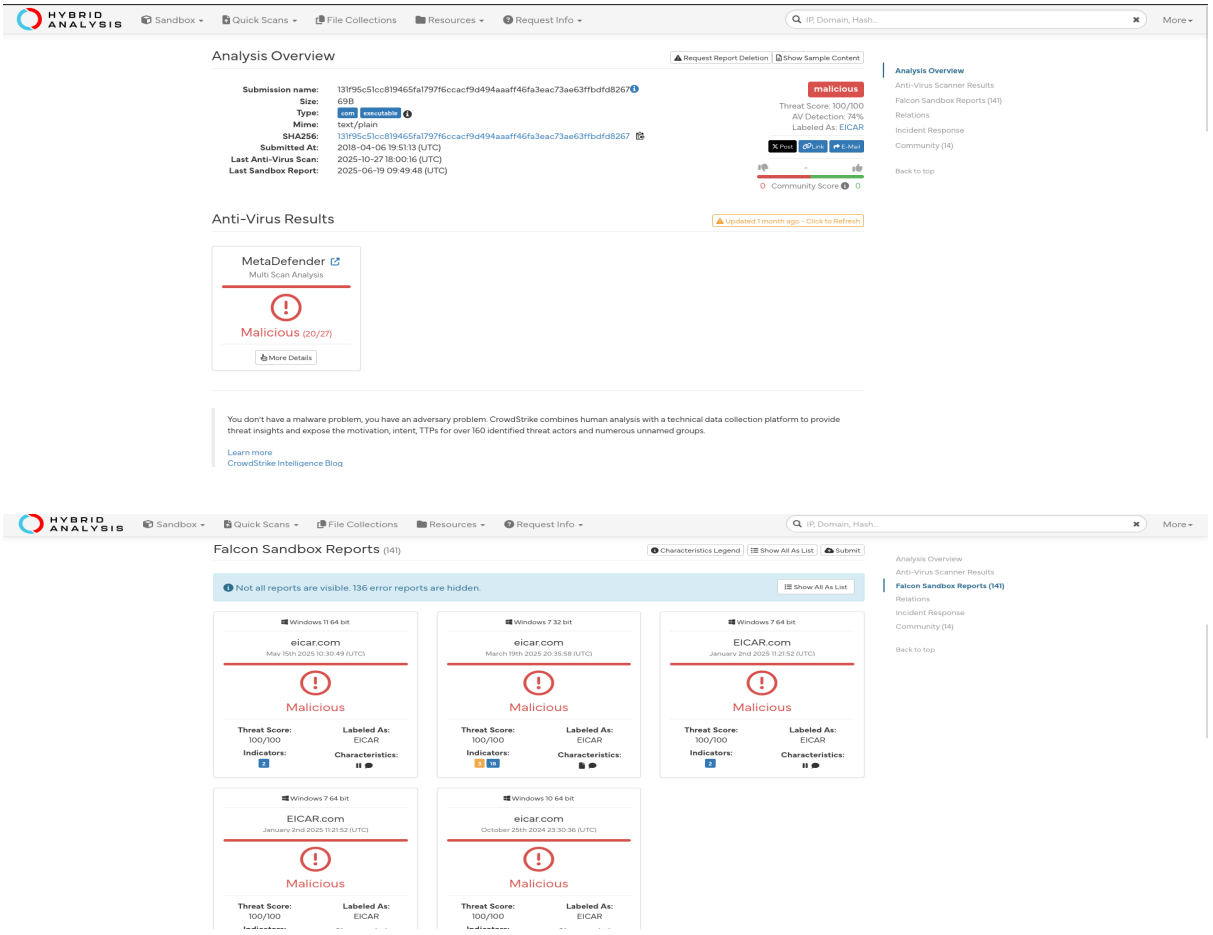
- **Hybrid Analysis**

Hybrid Analysis was used to observe the behavior of the EICAR test file in a sandboxed environment. The file was analyzed using a default Windows environment to simulate execution in a real system.

Observations

- The file was classified as **malicious (test file)**.
- Detection was based on signature recognition rather than runtime behavior.
- No file execution, system modification, or network activity was observed.

This confirms that the file is a non-functional test artifact and does not pose an actual security threat.



Hybrid Analysis overview and classification result

9.5 Sandbox Behavior Summary

The EICAR test file was analyzed using Hybrid Analysis to simulate malware detection. The sandbox environment correctly identified the file as malicious through signature-based detection. No runtime execution, system modification, or network activity was observed, confirming that the file is a safe test artifact used for validating antivirus detection mechanisms.

9.6 Security Impact and Analysis

The malware analysis task demonstrates the effectiveness of modern antivirus and sandbox-based detection systems in identifying known malicious signatures. While the EICAR file does not represent a real threat, this exercise highlights how quickly security tools can flag suspicious files. In real-world scenarios, similar detection mechanisms help prevent malware execution and reduce the impact of cyberattacks.

10. Password Security

10.1 Purpose of Password Security Assessment

Password security is a fundamental component of system protection, as weak or poorly managed passwords are one of the most common causes of unauthorized access. The purpose of this task was to demonstrate secure password management practices, generate strong passwords, and evaluate the effectiveness of authentication controls against weak password attempts in a controlled lab environment.

10.2 Tool Used – KeePassXC

KeePassXC@ is an open-source password manager that securely stores credentials in an encrypted database. It helps users generate and manage strong, unique passwords, reducing the risk of password reuse and brute-force attacks.

A new password vault was created and protected using a strong master password to demonstrate secure credential storage.

10.3 Secure Password Vault Creation

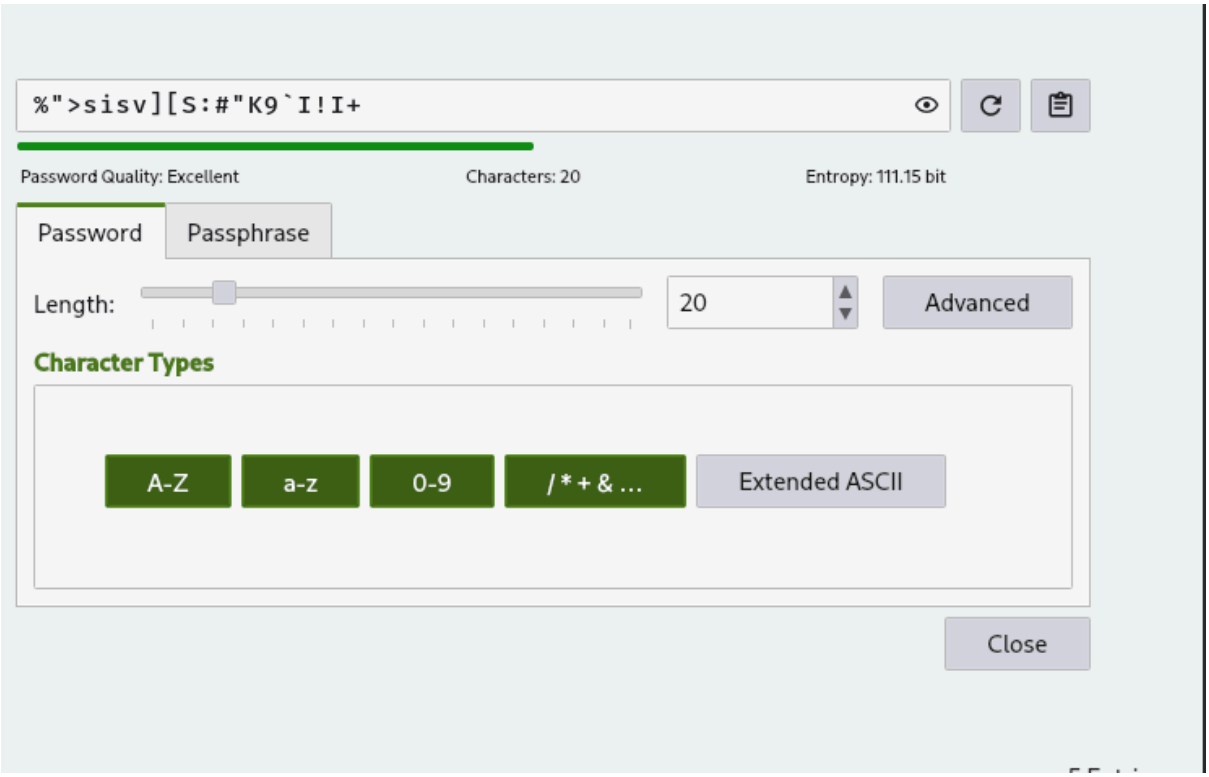
A new KeePassXC database named *Password_Security_Test* was created. The vault was protected with a strong master password, ensuring that all stored credentials remain encrypted and inaccessible without proper authorization. Default encryption settings were retained to maintain strong cryptographic protection.

10.4 Strong Password Generation

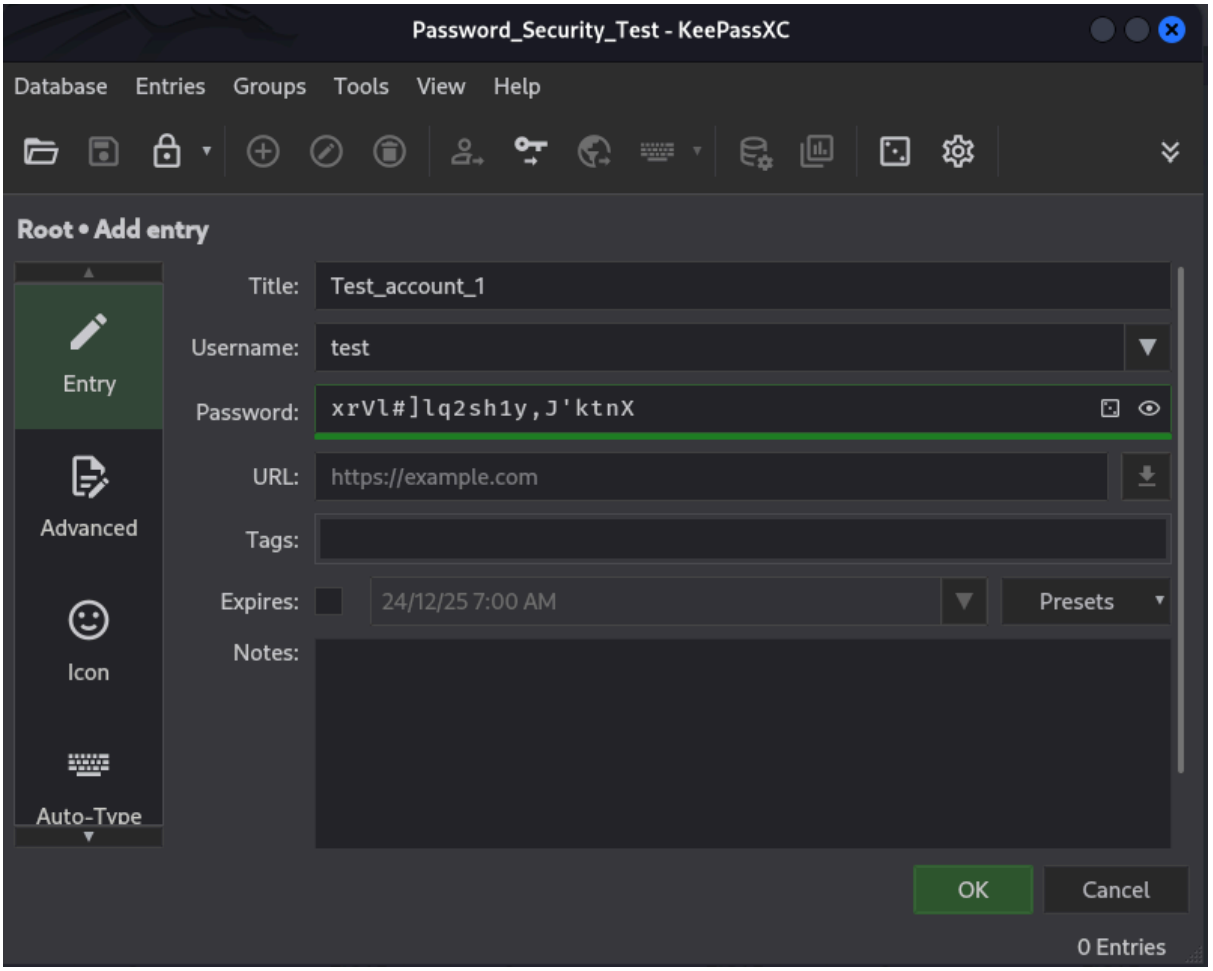
Using the built-in password generator in KeePassXC, five strong passwords were generated with the following characteristics:

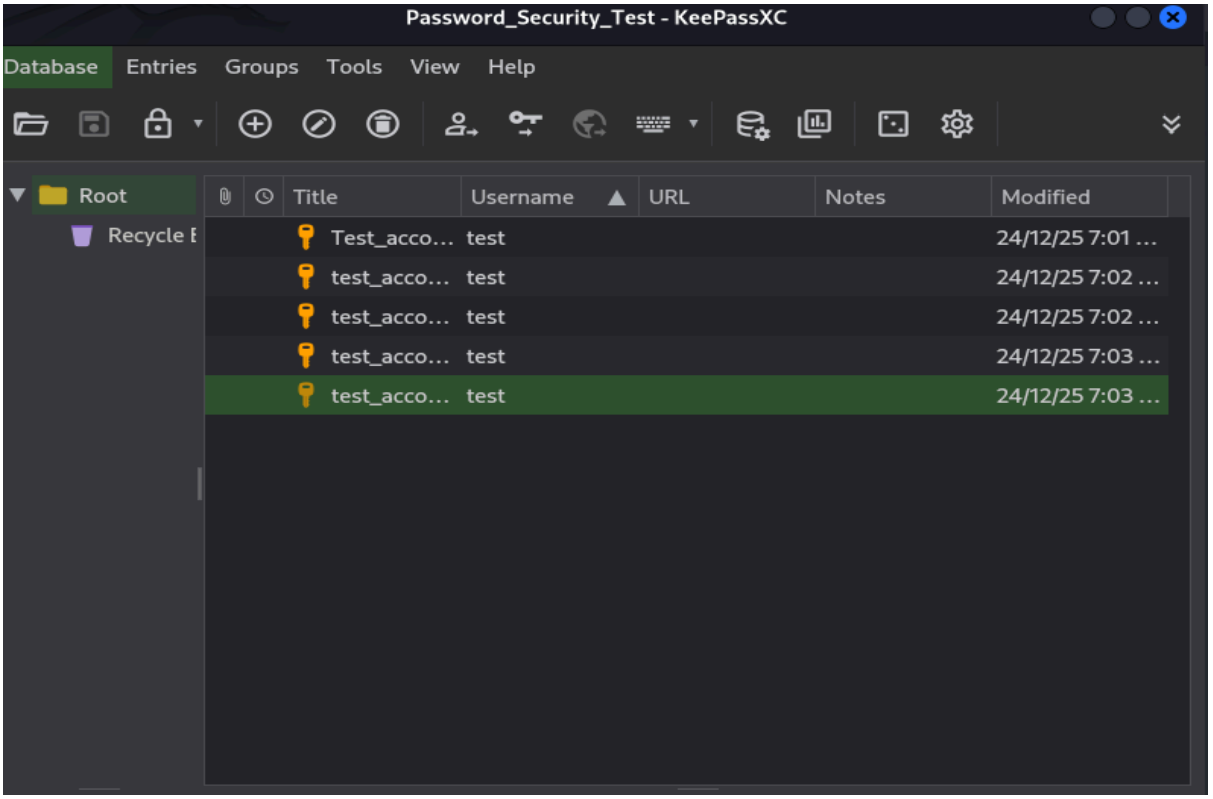
- Minimum length of **16 characters**
- Combination of **uppercase letters, lowercase letters, numbers, and special symbols**
- Randomized structure to prevent predictability

These passwords were stored securely in the password vault under test account entries.



KeePassXC password generator settings and generated passwords





Password entries saved inside the KeePassXC vault

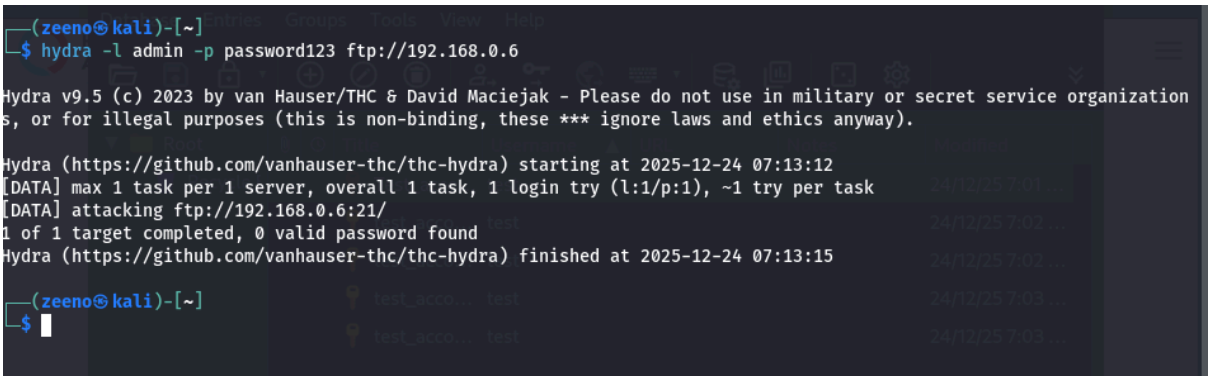
10.5 Weak Password Testing Using Hydra

To demonstrate the risks associated with weak passwords, a controlled password attack attempt was performed using **Hydra**, a commonly used password-testing tool. A single weak password (password123) was tested against the FTP service running on the target system. This test was intentionally limited to one password attempt to avoid brute-force behavior and ensure ethical testing.

Command Used:

hydra -l admin -p password123 ftp://192.168.0.6

The attack attempt failed, indicating that the FTP service does not accept common or easily guessable passwords. This result highlights the effectiveness of basic password security controls on the tested service.



Hydra execution output showing no valid password found

10.6 Analysis of Results

The failure of the weak password authentication attempt demonstrates the importance of enforcing strong password policies. While the password test did not succeed, it effectively illustrates how attackers commonly attempt access using widely known passwords. Proper password configuration and policy enforcement significantly reduce the risk of such attacks succeeding.

10.7 Security Impact

If weak passwords were allowed, attackers could easily gain unauthorized access using automated tools with minimal effort. Secure password management, combined with strong password policies and the use of password managers, plays a critical role in protecting systems from credential-based attacks.

11. MITRE ATT&CK Mapping

The exploitation activity performed during this assessment maps to **MITRE ATT&CK technique T1059 – Command and Scripting Interpreter**, as command execution was achieved through a remote shell after successful exploitation of a vulnerable service.

12. Attack Path Summary

Reconnaissance
↓
Network Scanning (Nmap)
↓
Vulnerability Assessment (OpenVAS)
↓
Exploitation (Metasploit)
↓
Post-Exploitation & Persistence

13. Recommendations

- Patch and update vulnerable services regularly.
- Disable unused and unnecessary network services.
- Implement strong password policies and multi-factor authentication.
- Conduct periodic vulnerability assessments and penetration testing.
- Monitor logs and network traffic for suspicious activity.

14. Conclusion

This Red Team assessment demonstrated how exposed services and unpatched vulnerabilities can lead to complete system compromise. The exercise provided hands-on experience with Red Team tools and methodologies while reinforcing the importance of proactive security measures. Implementing the recommended controls can significantly reduce the risk of successful cyberattacks.

15. References

- Nmap Official Documentation
- OpenVAS Documentation

- Metasploit Unleashed
- MITRE ATT&CK Framework
- SANS Red Team Resources