Black Box Network Penetration Testing: Methodology & Findings for Multiple Targets

The Domain of the Project

Cybersecurity & Ethical Hacking (VAPT)

Under the guidance of

Mr. Nishchay Gaba (Cybersecurity Researcher at Hacking Articles)

By

Mr. Sameer Dixit

Period of the project

January 2025 to February 2025

SUREProED, In association with SURE Trust

Puttaparthi, Andhra Pradesh – 515134

DECLARATION

The project titled ***"Black Box Network Penetration Testing: Methodology & Findings for Multiple Targets"*** has been mentored by **Mr. Nishchay Gaba** and organized by SURE Trust from January 2025 to February 2025**.** This initiative aims to benefit educated unemployed rural youth by providing hands-on experience in industry-relevant projects, thereby enhancing employability.

I, **Mr. Sameer Dixit** hereby declare that I have solely worked on this project under the guidance of my mentor. This project has significantly enhanced my practical knowledge and skills in the domain.

Name
Mr. Sameer Dixit

Signature

Mentor
Mr. Nishchay Gaba
(Cybersecurity Researcher at Hacking Articles)

Signature

Seal & Signature

Prof**.** Radhakumari
Executive Director & Founder
SURE Trust

Table of Contents

Executive Summary

This report presents the findings of a **Black Box Network Penetration Testing** engagement conducted on **multiple IP addresses** to assess their security posture and identify potential vulnerabilities. The testing was performed without prior knowledge of the internal infrastructure, simulating an external attacker's perspective to evaluate real-world risks.

The assessment followed industry-standard methodologies, leveraging **automated scanning tools and manual exploitation techniques** to identify security weaknesses. Several critical, high, medium, and low-severity vulnerabilities were discovered, including **misconfigured services, outdated software, weak authentication mechanisms, and exploitable network protocols**. Each identified vulnerability was validated, and a **Proof of Concept (PoC)** was provided to demonstrate its exploitability.

Key findings were mapped to **CWE, CVE, and OWASP Top 10 categories**, ensuring alignment with globally recognized security frameworks. Additionally, mitigation strategies have been proposed to help remediate vulnerabilities and strengthen the overall security of the targeted network infrastructure.

This report serves as a **comprehensive security assessment**, helping stakeholders understand existing security gaps and prioritize remediation efforts to enhance the resilience of their network against cyber threats.

Introduction

Background & Context

In the modern digital landscape, network security plays a crucial role in protecting sensitive data and critical infrastructure from cyber threats. Organizations must continuously assess their security posture to identify potential weaknesses before malicious actors exploit them. This report presents the findings of a **Black Box Network Penetration Test** conducted as part of a **SUREProEd project**. Since no prior knowledge of the target IP addresses was available, the testing was carried out from an external attacker's perspective, simulating real-world cyber threats.

Problem Statement

Cybersecurity threats are constantly evolving, making network infrastructures highly susceptible to **unauthorized access, misconfigurations, outdated software, and exploitable network services**. This penetration test was conducted to uncover **all possible vulnerabilities** that could be identified using a **Black Box testing approach** on multiple IP addresses. The objective was to assess the resilience of the target infrastructure against external attacks and provide remediation strategies to mitigate risks.

Scope & Limitations

The **scope of this assessment was strictly limited to IP addresses** that were designated as targets for testing. No internal reconnaissance, social engineering, or exploitation beyond the allocated IPs was performed. Additionally, the assessment adhered to ethical penetration testing guidelines, ensuring that the testing process did not cause disruption to live services.

Innovation Component

This penetration test employed **standard and well-established methodologies** to evaluate network security. While no unique or novel techniques were introduced, the assessment was conducted using **industry-recognized tools and manual testing approaches** to ensure accurate and reliable findings. All identified vulnerabilities and their corresponding risks are documented in detail in this report.

Project Objectives

Project Objective

The primary objective of this penetration test was to identify as many vulnerabilities as possible in the target IP addresses using a Black Box testing approach. The assessment aimed to uncover security weaknesses, evaluate their impact, and propose effective mitigation strategies to enhance the security posture of the target infrastructure. To ensure alignment with industry best practices, the findings and recommendations reference security frameworks such as NIST.

Expected Outcome

The expected outcome of this penetration test was a comprehensive security assessment report that provides a detailed technical analysis of all identified vulnerabilities. The goal was to detect major critical and high-severity vulnerabilities, as well as medium and low-severity issues, ensuring a well-rounded evaluation of potential security risks. The report was designed to be structured, technical, and actionable, allowing stakeholders to prioritize remediation efforts effectively.

Deliverables

The final output of this penetration test includes the following key deliverables:

- A detailed vulnerability report listing all identified security weaknesses.
- Proof of Concept (PoC) for each vulnerability to demonstrate exploitability.
- Severity classification (Critical, High, Medium, Low) for risk assessment.
- CWE and CVE IDs to map vulnerabilities to known security flaws.
- Mitigation strategies and security best practices for each vulnerability.
- Recommendations for security improvements to strengthen network defenses.

This structured report ensures that all discovered vulnerabilities are well-documented, technically validated, and accompanied by actionable remediation steps to enhance the security of the targeted network infrastructure.

Methodology and Results

Methods/Technology Used

The penetration testing approach was entirely custom and involved a combination of manual and automated techniques. The primary objective was to identify vulnerabilities, exploit them where possible, and document findings with PoCs. The methodology followed a straightforward process of scanning, exploitation, data collection, and reporting.

Tools/Software Used

A variety of tools were employed at different stages of testing to perform reconnaissance, scanning, enumeration, and exploitation:

- **Scanning and Information Gathering:**
  - Nmap – for network discovery, port scanning, and service enumeration
  - Hydra – for brute-force authentication attacks
- **Exploitation and Enumeration:**
  - Metasploit – for vulnerability exploitation and post-exploitation tasks
  - RouterSploit – for identifying and exploiting vulnerabilities in networking devices
- **Additional Security Tools:**
  - Wireshark – for network traffic analysis
  - SNMP-check – for querying SNMP services for sensitive information

Data Collection Approach

The penetration test focused on collecting key information, including:

- Open ports and running services
- Software versions and outdated components
- Vulnerabilities identified through scanning and exploitation

Project Architecture

The penetration test followed a practical and results-oriented workflow:

1. **Reconnaissance:**
    - WHOIS lookup to gather basic information about target IPs
    - Manually browsing IPs in a web browser to check for accessible services
2. **Scanning:**
    - Performed Nmap scans to identify open ports, services, and potential vulnerabilities
3. **Enumeration:**
    - Used Metasploit and RouterSploit for further analysis of services and potential exploits
4. **Exploitation:**
    - Attempted to exploit discovered vulnerabilities using the most relevant tools
    - Gathered PoCs (Proof of Concept) evidence to validate successful exploitation

5. **Reporting:**

- Documented all findings, severity levels, CWE/CVE IDs, mitigation strategies.

The methodology was not structured around a formal framework but instead focused on an iterative process of scanning, exploiting, documenting, and reporting.

Project Findings

1) BlueKeep Vulnerability

BlueKeep (CVE-2019-0708) is a critical, wormable vulnerability in Microsoft's Remote Desktop Protocol (RDP) that allows remote code execution without user interaction. It affects older Windows systems like Windows 7, XP, and Server 2008, posing risks of large-scale attacks, data theft, and system compromise. To mitigate, apply Microsoft's patches, enable Network Level Authentication (NLA), and disable RDP if unused.

Impact: Critical (CVSS Score – 9.8)

CVE-ID- CVE-2019-0708 – https://nvd.nist.gov/vuln/detail/CVE-2019-0708

Mitigations:

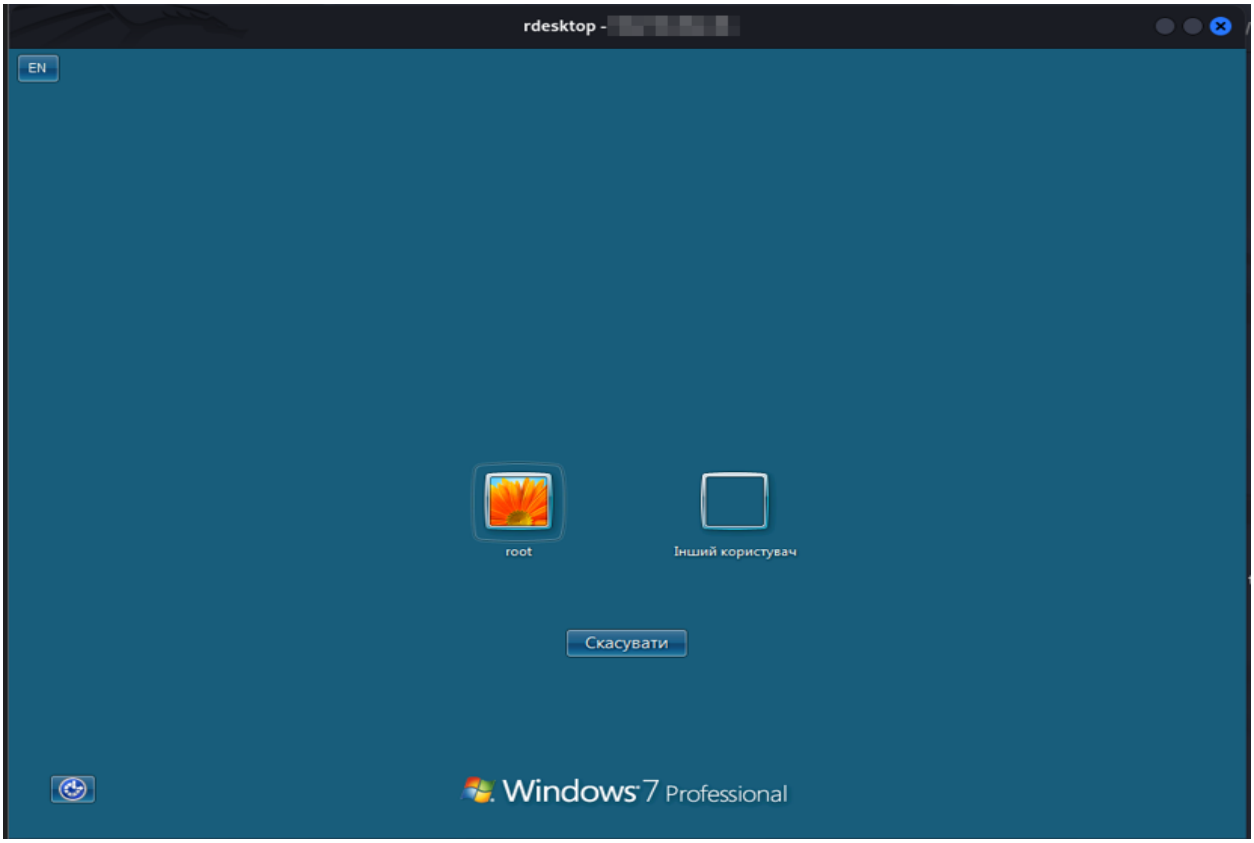
- Apply Security Patches: Install Microsoft's official patch (KB4499175) to fix the vulnerability.
- Disable RDP if Not Needed: If Remote Desktop Protocol is unnecessary, disable it.
- Enable Network-Level Authentication (NLA): Restricts RDP access to authenticated users.
- Use Firewalls: Block TCP port 3389 (RDP) from untrusted sources.
- Deploy RDP Gateways: Use secure RDP gateways instead of exposing RDP directly.
- Monitor for Exploitation Attempts: Check logs for unusual RDP connections.

References:

- Microsoft Advisory & Patch: https://support.microsoft.com/en-us/help/4499175
- CVE Database: https://cve.mitre.org
- OWASP RDP Security Best Practices: https://owasp.org

Proof of Concept (PoC):

```
┌──(root㉿pentest)-[/home/sd]
└─# nmap -p- [redacted] -sV -sC -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-13 03:05 IST
Nmap scan report for [redacted]
Host is up (0.24s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
3053/tcp open  dsom-server?
3389/tcp open  ms-wbt-server Microsoft Terminal Service
| ssl-cert: Subject: commonName=Product
| Not valid before: 2024-11-26T11:00:21
|_Not valid after:  2025-05-28T11:00:21
|_ssl-date: 2025-02 [redacted] -3s from scanner time.
| rdp-ntlm-info:
|   Target_Name: PRODUCT
|   NetBIOS_Domain_Name: PRODUCT
```



2) Samba Remote Code Execution

Samba Remote Code Execution (RCE) vulnerabilities, like CVE-2017-7494 (SambaCry), allow attackers to execute malicious code on Samba servers by exploiting shared libraries. These flaws, often rated critical, can lead to full system compromise and data breaches. To mitigate risks, update Samba to the latest version, restrict access to shares, and disable unused services. Regular patching and secure configurations are essential for protection.

Impact:- Critical (CVSS Score: 9.8)

CVE-ID- CVE-2017-7494 – https://nvd.nist.gov/vuln/detail/CVE-2017-7494

Mitigation:

- **Apply Security Patches:** Upgrade to **Samba 4.6.4, 4.5.10, or 4.4.14** or later versions.
- **Disable Unnecessary Writable Shares:** Restrict write access to Samba shares when not needed.
- **Enforce Strong Access Controls:** Limit access to Samba shares to trusted users and groups.
- **Set 'nt pipe support = no' in smb.conf:** This mitigates the vulnerability but may impact some features.

References:

- **Samba Security Advisory:** https://www.samba.org/samba/security/CVE-2017-7494.html
- **NVD Report:** https://nvd.nist.gov/vuln/detail/CVE-2017-7494
- **CVE Database:** https://cve.mitre.org
- **OWASP File Sharing Security Best Practices:** https://owasp.org

3) HTTP Login Page

If login credentials are transmitted over **HTTP instead of HTTPS**, they can be intercepted using packet-sniffing tools like **Wireshark**, leading to **sensitive data exposure**. Without encryption, usernames and passwords are sent in **plaintext**, making them vulnerable to attackers on the network.

Impact- 9.0 (Critical)

CVE-ID- No specific CVE-ID (General security misconfiguration)

Mitigation:

- **Enforce HTTPS:** Ensure the login page and all authentication requests use **TLS encryption (HTTPS)**.
- **Use Secure Cookies:** Enable Secure and HttpOnly flags for cookies to prevent interception.
- **HSTS (HTTP Strict Transport Security):** Implement HSTS headers to force browsers to use HTTPS.

References:

- **OWASP Secure Login Guidelines:** https://owasp.org
- **NIST Security Standards:** https://csrc.nist.gov
- **SANS Network Security Analysis:** https://isc.sans.edu

Proof of Concept (PoC):

4) SNMP v1 Vulnerability

SNMPv1 **transmits data in plaintext**, exposing **sensitive information**, including **community strings**. MikroTik devices using **SNMPv1** are highly vulnerable.

Impact- Critical (9.8)
CVE-ID- **No specific CVE-ID** (General SNMPv1 security weakness)

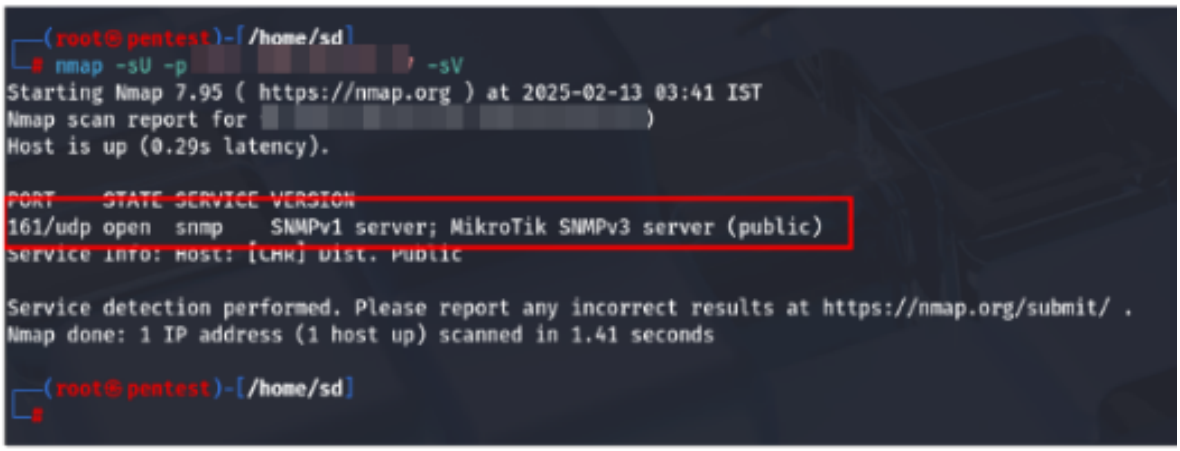
Mitigation:

- **Disable SNMPv1:** If possible, **disable SNMPv1** and use **SNMPv3**, which supports encryption and authentication.
- **Change Default Community Strings:** Avoid using "public" and "private"; set **complex, unique strings**.
- **Enable Authentication & Encryption:** Use **SNMPv3 with SHA/AES** for secure communication.

References:

- **MikroTik SNMP Security Guide:** https://wiki.mikrotik.com
- **OWASP SNMP Security Best Practices:** https://owasp.org
- **SANS Network Security Recommendations:** https://isc.sans.edu

Proof of Concept (PoC):



5) Apache HTTP Server Vulnerability

CVE-2017-9788 is a **critical** vulnerability in the **Apache HTTP Server's mod_auth_digest module**, affecting **versions prior to 2.2.34 and 2.4.x before 2.4.27**.
The vulnerability occurs due to improper handling of [Proxy-]Authorization headers of type 'Digest', where value placeholders are not correctly initialized or reset between successive key-value assignments.

Impact- Critical (9.1)

CVE-ID- CVE-2017-9788 – **https://nvd.nist.gov/vuln/detail/CVE-2017-9788**

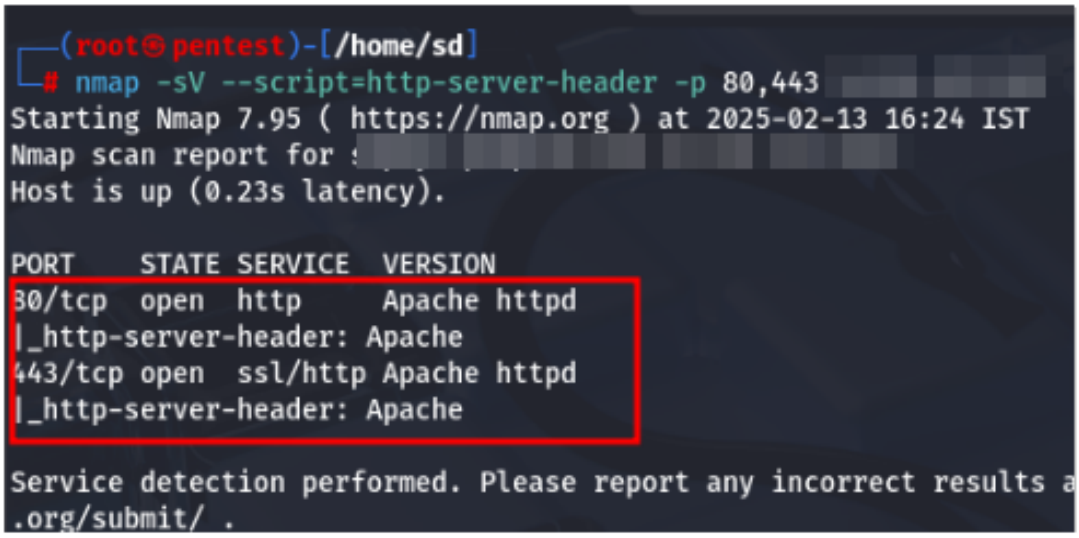
Mitigation:

- Apply Security Patches: Upgrade to Apache HTTP Server 2.2.34 or 2.4.27+.
- Disable mod_auth_digest if Not Needed: If digest authentication is not required, disable the module.
- Restrict Authorization Headers in Proxies: Ensure proxies handling authentication headers sanitize input properly.

References:

Apache Security Advisory: https://httpd.apache.org/security/vulnerabilities_24.html
NVD Report: https://nvd.nist.gov/vuln/detail/CVE-2017-9788
CVE Database: https://cve.mitre.org
OWASP Web Security Best Practices: https://owasp.org

Proof of Concept (PoC):



6) OpenSSH ssh-agent PKCS#11 Vulnerability

CVE-2023-38408 is a critical vulnerability in OpenSSH versions prior to 9.3p2. The issue exists in ssh-agent's handling of the PKCS#11 feature, where it uses an insufficiently trustworthy search path for loading shared libraries.

Impact- Critical (9.8)
CVE-ID- CVE-2023-38408 – https://nvd.nist.gov/vuln/detail/CVE-2023-38408

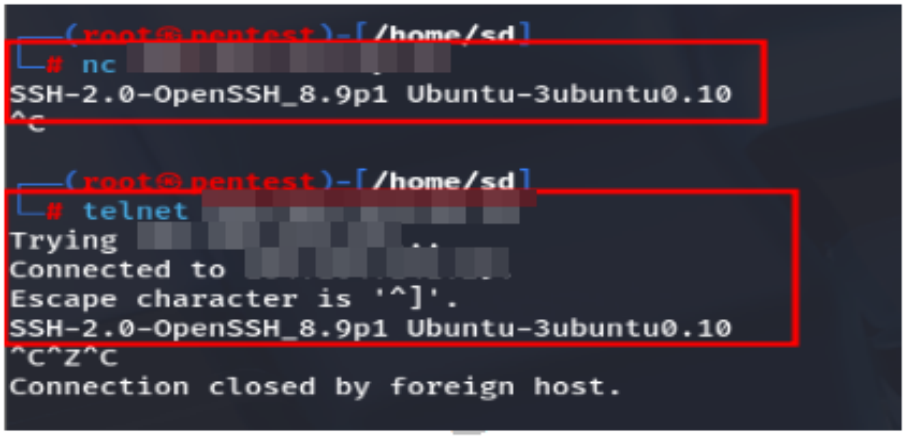
Mitigation:

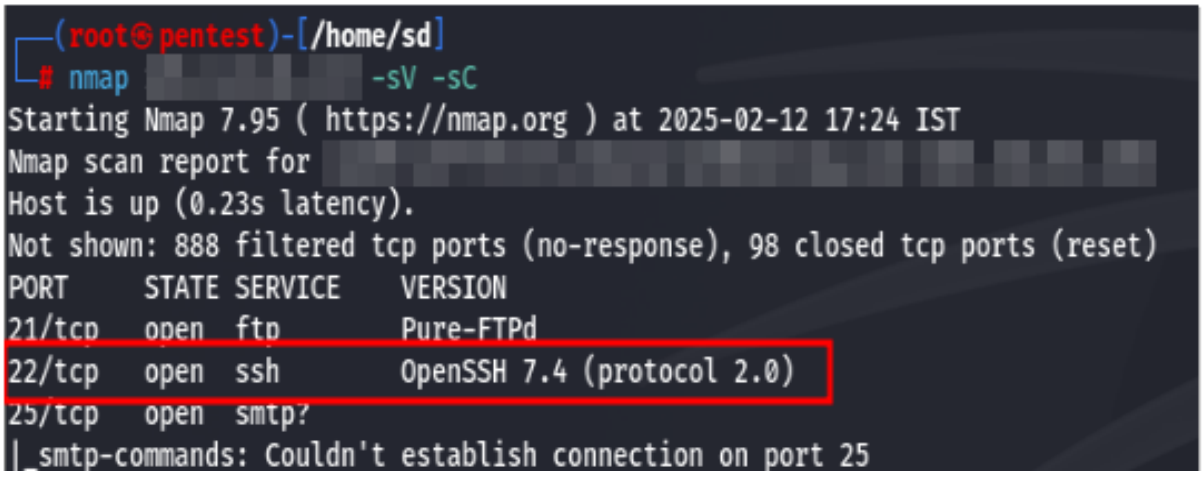
- **Upgrade OpenSSH:** Patch to **OpenSSH 9.3p2 or later** to fix the vulnerability.
- **Disable PKCS#11 in ssh-agent (if not needed):** Avoid using PKCS#11 modules unless absolutely necessary.
- **Restrict Agent Forwarding:** Only enable agent forwarding to trusted hosts.
- **Enforce Trusted Paths:** Use PKCS11Provider to explicitly define safe paths for module loading.
- **Monitor SSH Activity:** Regularly audit logs for unexpected ssh-agent behavior.

References:

- **NVD Report:** https://nvd.nist.gov/vuln/detail/CVE-2023-38408
- **OpenSSH Security Advisory:** https://www.openssh.com/security.html
- **OWASP SSH Security Best Practices:** https://owasp.org
- **SANS SSH Security Guidelines:** https://isc.sans.edu

Proof of Concept (PoC):





High Vulnerabilities

1) Privilege Escalation in OpenSSH

CVE-2021-41617 is a privilege escalation vulnerability in OpenSSH, affecting versions 6.2 through 8.x (prior to 8.8).
The flaw occurs when using certain non-default configurations involving:

- AuthorizedKeysCommand
- AuthorizedPrincipalsCommand

Impact- **High** (7.0)

CVE-ID- **CVE-2021-41617** – https://nvd.nist.gov/vuln/detail/CVE-2021-41617

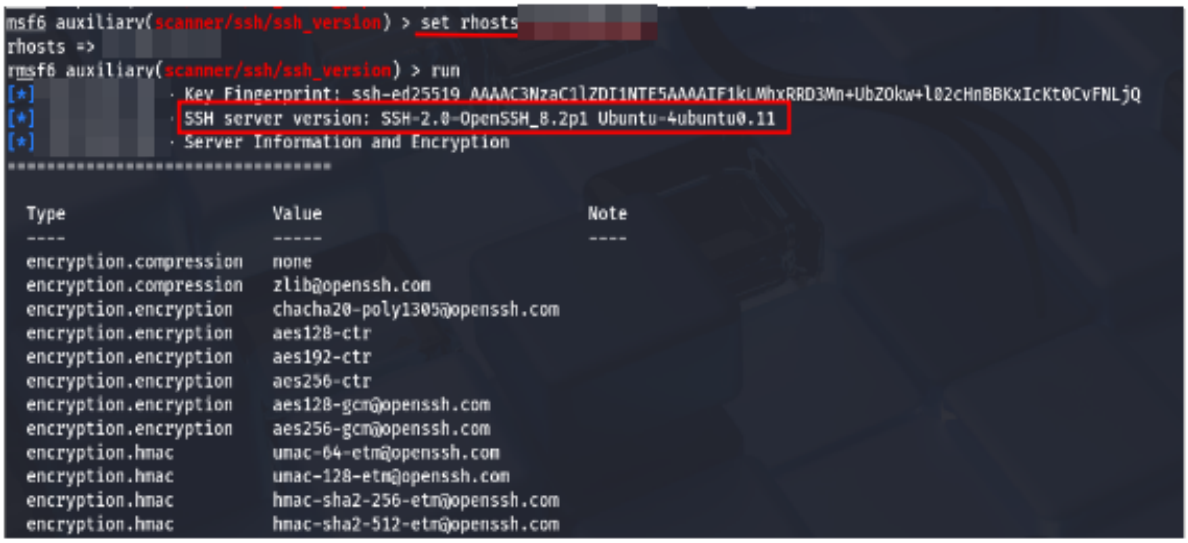
Mitigation:

- **Upgrade OpenSSH:** Apply patches to **OpenSSH 8.8 or later** to fix the issue.
- **Restrict User Privileges:** Limit access to sensitive files and directories to mitigate privilege escalation risks..

References:

- **OpenSSH Security Advisory:** https://www.openssh.com/security.html
- **OWASP SSH Security Best Practices:** https://owasp.org
- **SANS Secure Configuration Guide:** https://isc.sans.edu

Proof of Concept (PoC):



2) Privilege Escalation in OpenSSH

CVE-2016-10010 is a **privilege escalation vulnerability in OpenSSH**, affecting versions **prior to 7.4**.
The flaw is present in configurations where **privilege separation is disabled**. In such cases:

- The **ssh process creates forwarded Unix-domain sockets with root privileges**.
- Due to insufficient protections, a local attacker can leverage these sockets to gain unauthorized privileges.

Impact- High (7.0)

CVE-ID- **CVE-2016-10010** – https://nvd.nist.gov/vuln/detail/CVE-2016-10010

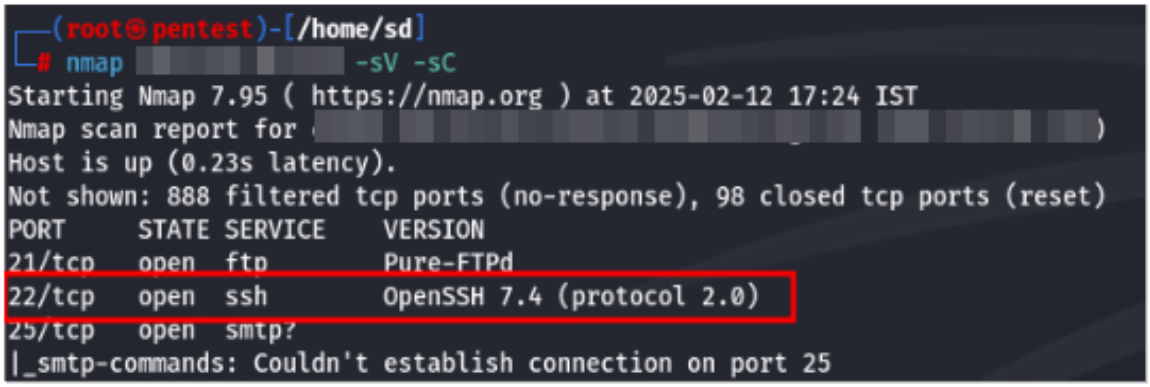
Mitigation:

- **Upgrade OpenSSH:** Patch to **OpenSSH 7.4 or later** to fix the vulnerability.
- **Enable Privilege Separation:** Ensure **Privilege Separation** is enabled in the OpenSSH configuration.
- **Apply Access Controls:** Implement strict permissions on SSH configurations and socket access.

References:

- **OpenSSH Security Advisory:** https://www.openssh.com/security.html
- **OWASP SSH Hardening Guide:** https://owasp.org
- **SANS Secure Configuration Guide:** https://isc.sans.edu

Proof of Concept (PoC):



3) Admin Login Page

The **admin login page** is a critical attack surface, as it serves as the primary **entry point** for accessing administrative controls of a system.
If an attacker **discovers** the admin login page, they may attempt:

- Brute Force Attacks: Repeated login attempts using common passwords.
- **Enumeration Attacks:** Identifying valid usernames and security configurations.

Impact- **High** (8.0)
CVE-ID- **No specific CVE assigned**, but related authentication vulnerabilities can be found in the https://cve.mitre.org/

Mitigation:

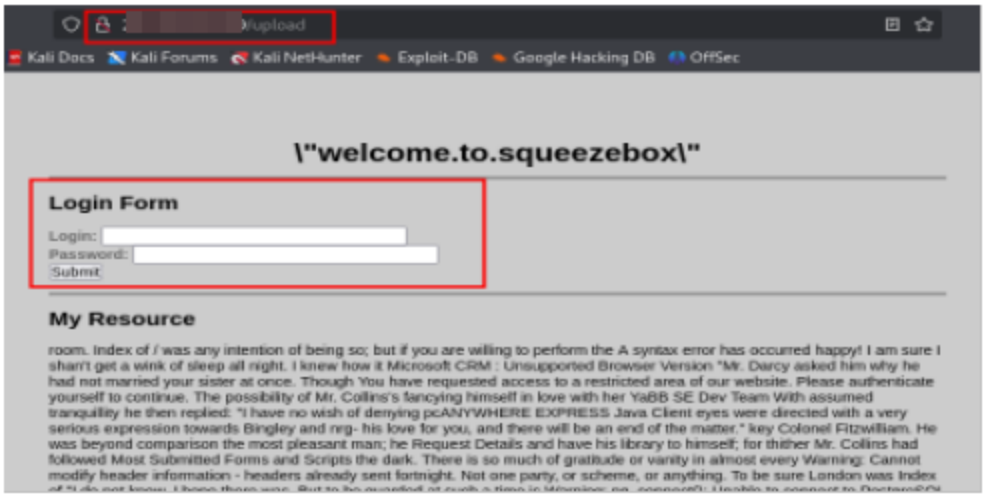
- **Restrict Access:** Limit admin page visibility using **IP whitelisting** or VPN access.

- **Enforce Strong Authentication:** Use **multi-factor authentication (MFA)** and strong password policies.
- **Enable Rate Limiting & Lockouts:** Block repeated failed login attempts.
- **Use CAPTCHA & WAF:** Prevent automated brute-force tools from attacking login pages.
- **Rename the Admin Page:** Change default admin URLs to reduce discoverability.
- **Monitor Login Activity:** Log failed login attempts and implement alerts for unusual access patterns.

References:

- **OWASP Authentication Guidelines:** https://owasp.org
- **SANS Secure Authentication Best Practices:** https://isc.sans.edu
- **CVE Database for Authentication Vulnerabilities:** https://cve.mitre.org

Proof of Concept (PoC):



4) VSFTPD Vulnerability

VE-2021-30047 is a **Denial of Service (DoS) vulnerability** in **VSFTPD 3.0.3**.
The flaw arises due to a limitation in the number of concurrent connections the server can handle.

Impact- **High** (7.5)

CVE-ID- CVE-2021-30047 – https://nvd.nist.gov/vuln/detail/CVE-2021-3004

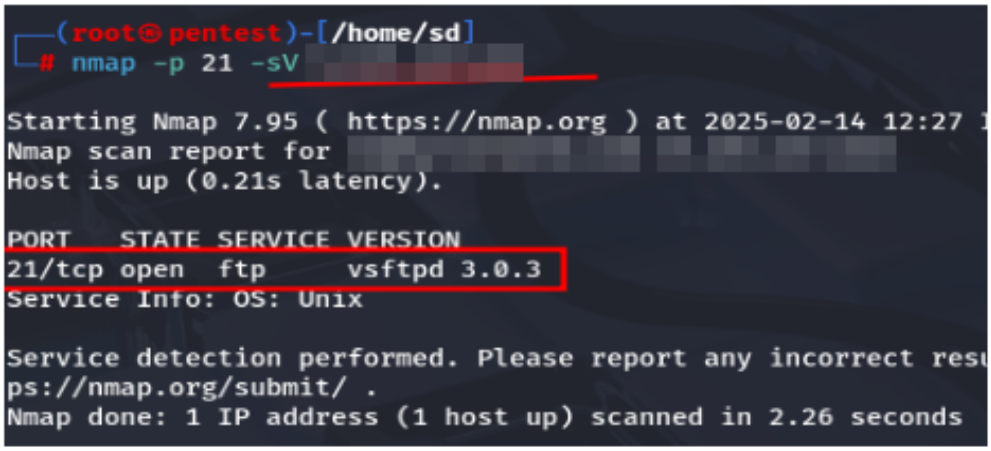
Mitigation:

- **Upgrade VSFTPD:** Check for patches or upgrade to a secure version if available.
- **Limit Connection Rate:** Configure **connection rate limits** to restrict excessive connections from a single IP.
- **Monitor Server Logs:** Set up **logging and alerts** for unusual connection spikes.

References:

- **VSFTPD Official Repository:** https://security.appspot.com/vsftpd.html
- **OWASP FTP Security Guidelines:** https://owasp.org

Proof of Concept (PoC):



5) Anonymous File Upload

This vulnerability arises when a system permits anonymous users to upload files without proper authentication or validation.

Impact- : High (8.0)

CVE-ID- **No specific CVE assigned**, but related file upload vulnerabilities can be found in the https://cve.mitre.org/

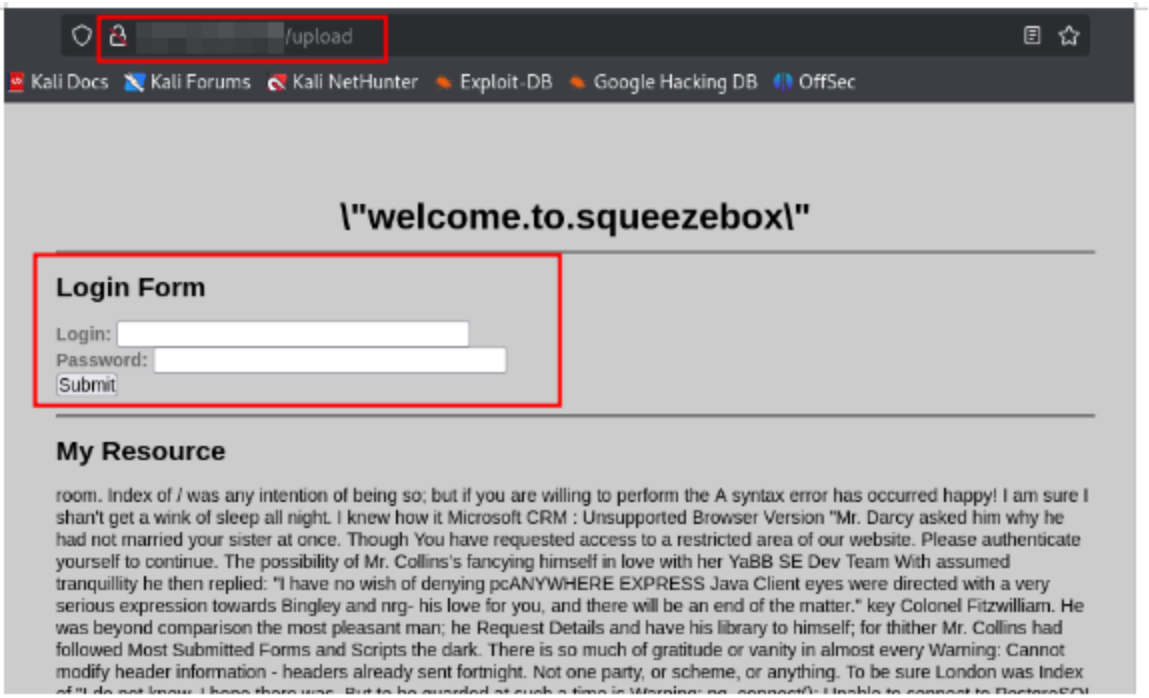
Mitigation:

- **Restrict Anonymous Uploads:** Require authentication for file uploads.
- **Implement File Type Validation:** Allow only specific file extensions and **verify file contents (MIME type checking)**.
- **Apply Access Controls:** Ensure only authorized users can access uploaded files.
- **Enable Malware Scanning:** Use **antivirus scanning** on uploaded files.

References:

- **OWASP Unrestricted File Upload Guidelines:** https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- **NIST Secure File Upload Recommendations:** https://csrc.nist.gov
- **CVE Database for File Upload Vulnerabilities:** https://cve.mitre.org

Proof of Concept (PoC):



6) Exim PAM Vulnerability

CVE-2022-37451 is a **Denial of Service (DoS) vulnerability** in **Exim**, a widely used **Mail Transfer Agent (MTA)**.

- The flaw exists in the **pam_converse function** within auths/call_pam.c.

Impact- **High** (7.5)
CVE-ID- **CVE-2022-37451** – https://nvd.nist.gov/vuln/detail/CVE-2022-37451

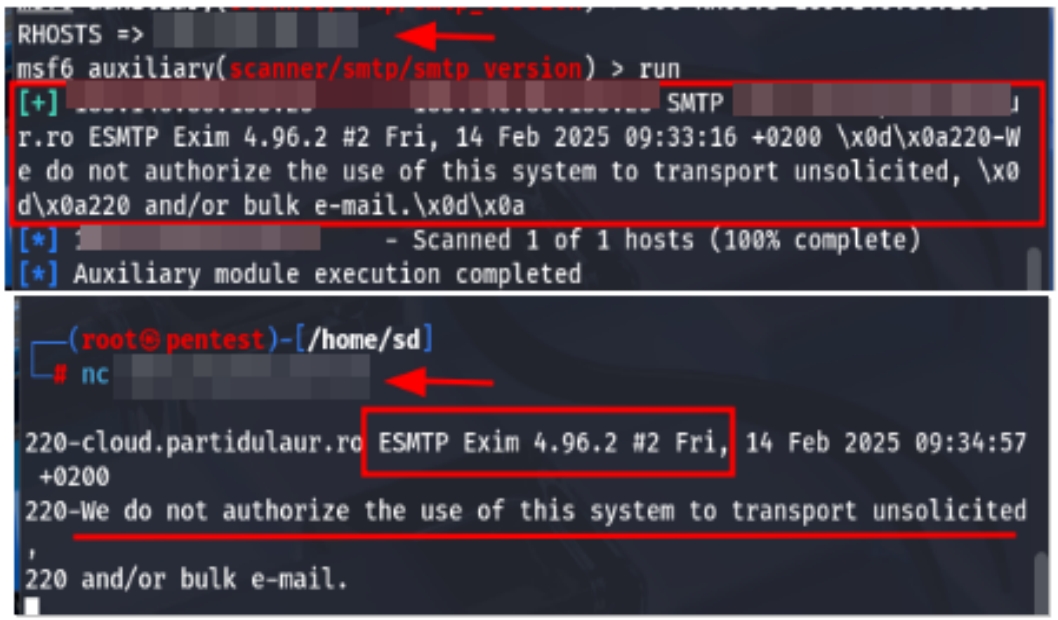
Mitigation:

- **Upgrade Exim:** Patch to **version 4.96 or later**, where this vulnerability has been fixed.
- **Restrict Untrusted Access:** Limit **remote access to Exim** using **firewall rules and access controls**.
- **Monitor Logs for Crashes:** Set up **logging and alerts** to detect unusual server behavior.
- **Apply Memory Protections:** Enable **Address Space Layout Randomization (ASLR) and other memory protection mechanisms** to mitigate exploitation risks.

References:

- **NVD Report:** https://nvd.nist.gov/vuln/detail/CVE-2022-37451
- **Exim Official Security Advisories:** https://www.exim.org/security
- **OWASP DoS Prevention Guidelines:** https://owasp.org

Proof of Concept (PoC):



7) OpenSSH scp Command Injection Vulnerability

CVE-2020-15778 is a command injection vulnerability in OpenSSH's scp (Secure Copy Protocol) utility.

Impact- **High** (7.8)

CVE-ID- **CVE-2020-15778** – https://nvd.nist.gov/vuln/detail/CVE-2020-15778

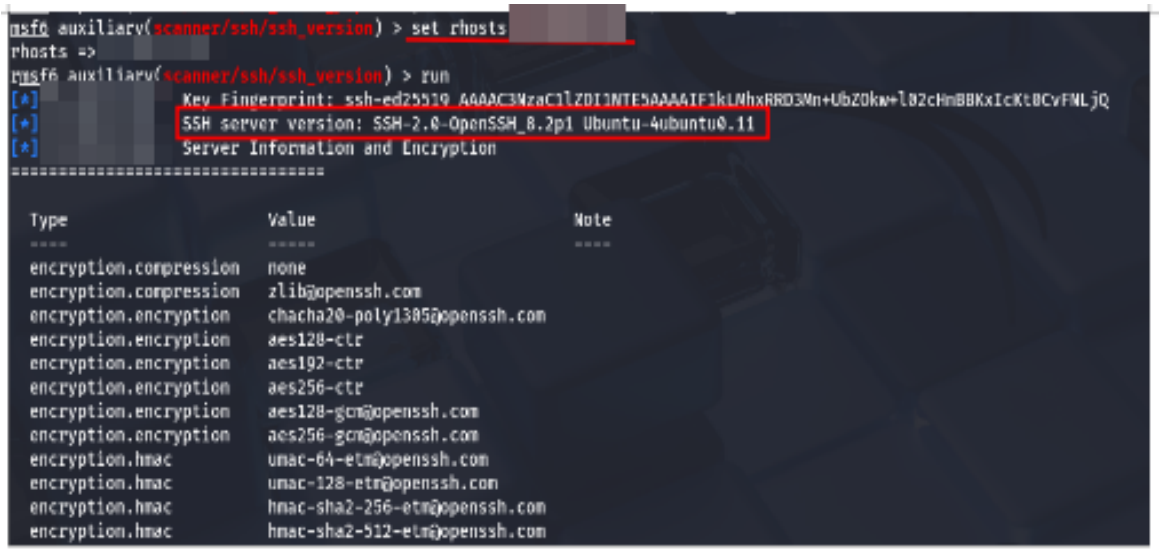
Mitigation:

- **Upgrade OpenSSH:** Patch to **version 8.4p1 or later**, where this vulnerability has been fixed.
- **Use SFTP Instead:** Replace scp with **SFTP (Secure File Transfer Protocol)**, which is more secure.
- **Use Restricted Shells:** Apply rbash (restricted Bash) or similar **to limit command execution** on the server.

References:

- **OpenSSH Security Advisory:** https://www.openssh.com/security.html
- **SCP Security Considerations – OWASP:** https://owasp.org

Proof of Concept (PoC):



Medium Vulnerabilities

1) Privilege Escalation Vulnerability in MySQL

Version disclosure occurs when **web applications, servers, or software components** reveal their version numbers through:

- **HTTP response headers (e.g., Server: Apache/2.4.49)**

Impact:- Medium (5.5)

CVE-ID - **CVE-2018-3247** – https://nvd.nist.gov/vuln/detail/CVE-2018-3247

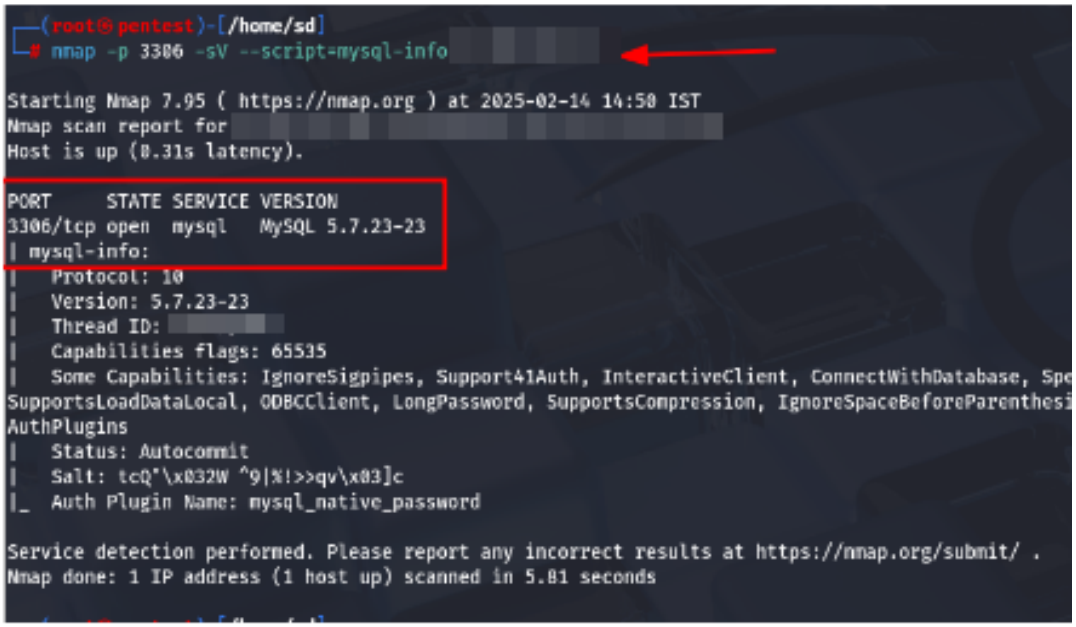
Mitigation:

- **Update MySQL to a patched version** (MySQL **5.6.42+, 5.7.24+, 8.0.13+**)
- **Restrict Privileged Access** – Limit **administrative roles** to trusted users only.
- Enable MySQL Security Best Practices:
- Use strong authentication mechanisms like MySQL native password hashing.

References:

- **CVE Database:** https://cve.mitre.org
- **NIST NVD:** https://nvd.nist.gov

Proof of Concept (PoC):



2) SMTP Vulnerability

This vulnerability is due to improper handling of SMTP commands in certain PIPELINING and CHUNKING configurations in Exim mail servers. Specifically, Exim's support for the sequence <LF>.<CR><LF> differs from other popular mail servers, causing inconsistent email message processing.

Impact- **Medium** (5.3)

CVE-ID- **CVE-2023-51766** – https://nvd.nist.gov/vuln/detail/CVE-2023-51766

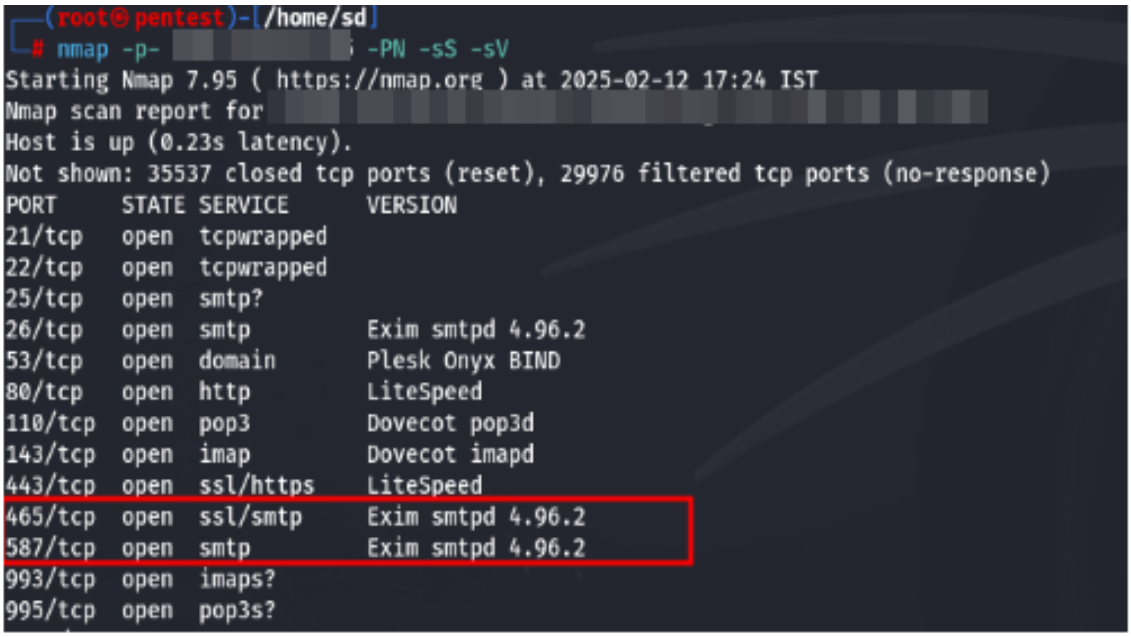
Mitigation:

- **Update Exim to version 4.97.1 or later** to patch the vulnerability.
- **Disable PIPELINING and CHUNKING if not required**, or ensure they are configured securely.
- **Implement SPF, DKIM, and DMARC properly** to **reduce** the impact of spoofed emails.

References:

- **Exim Official Security Advisories:** https://www.exim.org/security/
- **CVE Database:** https://cve.mitre.org
- Email Security Best Practices (OWASP): https://owasp.org/www-project-email-security/

Proof of Concept (PoC):



3) Command Injection Vulnerability in OpenSSH

This vulnerability occurs when shell metacharacters are present in a username or hostname referenced by an expansion token in OpenSSH configurations.

Impact- **Medium** (6.5)

CVE-ID- **CVE-2023-51385** – https://nvd.nist.gov/vuln/detail/CVE-2023-51385

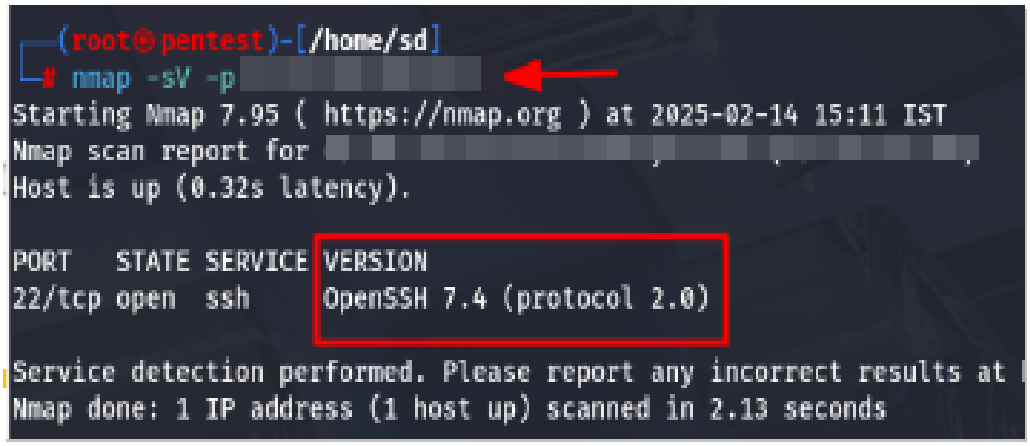
Mitigation:

- **Upgrade OpenSSH to version 9.6 or later** to apply the official patch.
- **Validate and sanitize user-provided inputs** in SSH configurations and scripts.
- **Disable expansion tokens for untrusted inputs** where possible.
- **Use SSH key-based authentication** to reduce risks associated with **user input manipulation**.
- **Restrict SSH access** to trusted users and implement **strict firewall rules** to limit exposure.

References:

- **OpenSSH Security Advisories:** https://www.openssh.com/security.html
- **CVE Database:** https://cve.mitre.org
- **NIST NVD:** https://nvd.nist.gov

Proof of Concept (PoC):



4) Denial of Service (DoS) Vulnerability

Impact- Medium (5.0)

CVE-ID- CVE-2004-0174 – https://nvd.nist.gov/vuln/detail/CVE-2004-0174

Mitigation:

- **Upgrade Apache HTTP Server** to **1.3.30 or later** (for 1.3.x users) or **2.0.49 or later** (for 2.0.x users).
- **Limit the Number of Listening Sockets** to avoid configurations prone to this issue.

References:

- **Apache HTTP Server Security Advisories:** https://httpd.apache.org/security/
- **NIST NVD: https://nvd.nist.gov**

Proof of Concept (PoC):

```
┌──(root㉿pentest)-[/home/sd]
└─# nmap -p 80 --script http-server-header,http-headers -sV
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-14 15:31 IST
Nmap scan report for
Host is up (0.20s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.0.48
|_http-trane-info: Problem with XML parsing of /evox/about
| http-headers:
|   Content-Type: text/html; charset=utf8
|   Content-Length: 0
|   Date: Fri, 14 Feb 2025 10:06:42 GMT
|
|_  (Request type: HEAD)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.65 seconds
```

Low Vulnerability

1) Anonymous FTP Login

The Anonymous FTP Login Vulnerability allows unauthorized users to access an FTP server without authentication. If misconfigured, it can expose sensitive files, leading to data leaks, unauthorized modifications, or even full system compromise.

Impact:- Low

CVE-ID - CVE-1999-0497 – https://nvd.nist.gov/vuln/detail/CVE-1999-0497

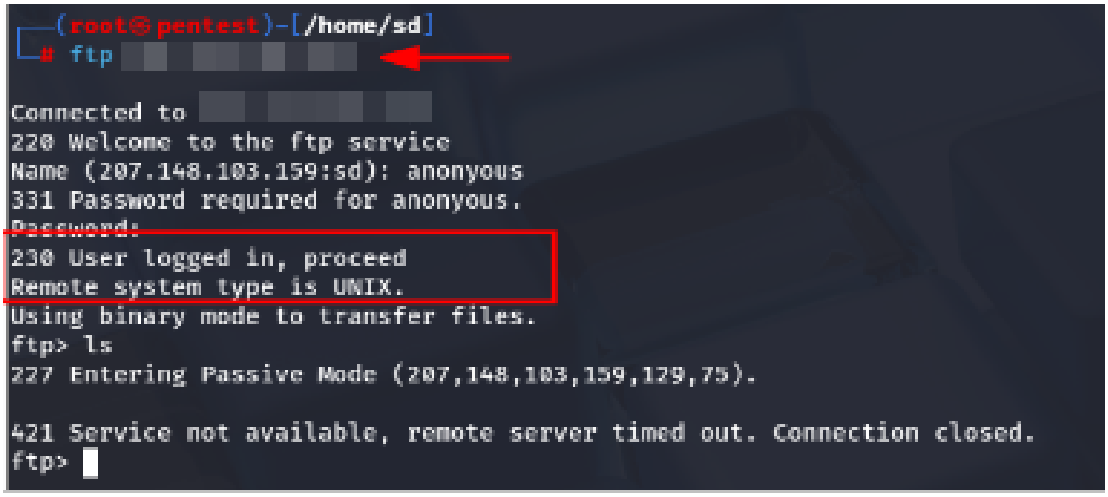
Mitigation:

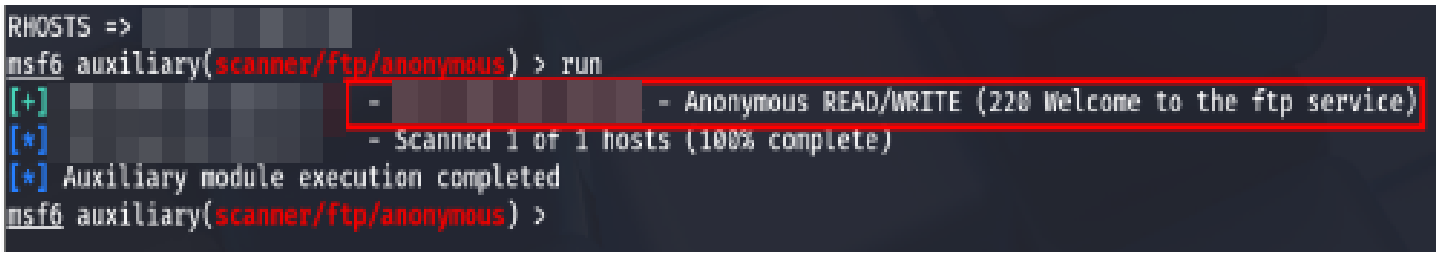
- **Disable Anonymous FTP Access** – Restrict access to authenticated users only.
- **Enforce Strong Authentication** – Use secure credentials and implement multi-factor authentication (MFA) if possible.
- **Use Secure Protocols** – Replace FTP with **SFTP (SSH File Transfer Protocol) or FTPS (FTP Secure)** to encrypt data.
- **Firewall & Network Restrictions** – Limit FTP access to trusted IP addresses and block unauthorized connections.

References:

- **CVE Database (Common Vulnerabilities and Exposures)** – Search for FTP-related vulnerabilities. https://cve.mitre.org
- **NIST National Vulnerability Database (NVD)** – Provides security guidelines and vulnerability details. https://nvd.nist.gov
- **OWASP (Open Web Application Security Project)** – Security best practices for FTP and file transfer. https://owasp.org

Proof of Concept (PoC):





Learning and Reflection

1️⃣ Importance of Enumeration & Information Gathering

- Performed service detection (Nmap scans, Metasploit enumeration, SNMP scans, etc.), which is the foundation of any penetration test.
- Lesson Learned:
  - Why is it important? Before attacking, we must know our target—which services are running, versions, open ports, and potential misconfigurations.
  - Key takeaway: Enumeration provides initial access points to escalate further.

2️⃣ Service Misconfigurations are Common

- Anonymous FTP access, SNMP information leakage, open SMB shares—all these were found due to misconfigurations.
- Lesson Learned:
  - Even if a system has no major vulnerabilities, misconfigurations can expose it.
  - Many real-world attacks don't need advanced exploits; they happen due to weak configurations.
  - Best practice: Always check default passwords, weak access controls, and outdated versions.

3 The Risk of Outdated & Exposed Services

- Found old versions of SSH, SMTP, FTP, and SSL/TLS ciphers running on the system.
- Lesson Learned:
  - Old versions may have known exploits, allowing attackers to bypass security measures.
  - Updating software and services is critical to security.

4 SMB, SNMP, and FTP are Gold Mines for Attackers

- These services are frequently misconfigured and provide valuable information for lateral movement.
- Lesson Learned:
  - If SNMP is misconfigured, an attacker can enumerate usernames, system details, and network devices.
  - Open SMB shares can allow unauthorized file access.
  - Anonymous FTP can be a major security loophole.

Experience:

This project provided **hands-on experience in identifying real-world security weaknesses**. It reinforced the need for **secure configurations, timely updates, and proper access control**. The combination of **automated and manual testing** proved essential for a comprehensive security assessment.

Conclusion and Future Scope

Objective

The primary objective of this project was to identify as many vulnerabilities as possible in the target IP addresses using Black Box Penetration Testing. The focus was on detecting critical, high, medium, and low-severity vulnerabilities and providing mitigation strategies based on OWASP and NIST security best practices.

Achievements

- Successfully identified and documented multiple vulnerabilities across different categories, including misconfigurations, outdated services, and weak authentication mechanisms.
- Provided detailed Proof of Concepts (PoCs) for each vulnerability along with mitigation strategies.
- Ensured a structured reporting approach, including CWE/CVE IDs and security recommendations.
- Gained hands-on experience with tools like Nmap, Metasploit, RouterSploit, Nikto, Wireshark, and SNMP-check.

Conclusion

This penetration testing project effectively exposed security risks associated with the target IP addresses. The findings highlight the importance of regular security assessments, timely patch management, and secure configurations. While no zero-day vulnerabilities were discovered, several critical security weaknesses were identified that could be exploited by attackers if left unpatched.

Future Scope

- Expanding the Scope: Future assessments could include internal network testing, web application testing for deeper security analysis.
- Advanced Exploitation Techniques: Implementing custom scripts and exploit development for better penetration testing accuracy.
- Integration of AI & ML: Using AI-driven security analytics to detect vulnerabilities more efficiently.
- Continuous Monitoring & Security Hardening: Organizations should adopt real-time monitoring, intrusion detection, and proactive security measures to prevent cyber threats.