



---

**School of Computer Science, Engineering and Applications(SCSEA)**

**Academic Year: 2025-26**

**MCA S.Y. (Sem III)**

**Subject : Ethical Hacking & Digital Forensics (CS2001)**

**Name of the Student: Jayesh Bhushan Bhangale**

**PRN: 20240804069**

**Title of Practical : Reconnaissance & Footprinting**

---

**AIM :** To perform reconnaissance and footprinting on a selected target using Whois lookup, DNS enumeration, Google Dorks, and OSINT tools such as Maltego or Shodan.

**THEORY:** Reconnaissance is the first step of ethical hacking. It involves passively or actively collecting information about a target before performing any attack simulation. This information helps in identifying network weaknesses.

Main techniques include:

- Whois Lookup
- DNS Enumeration
- Email discovery
- Metadata extraction
- Google Dorking



## **PERFORM OPERATION:**

### **1. Perform Whois Lookup**

- Open any Whois lookup website such as [whois.domaintools.com](http://whois.domaintools.com) or [whois.com](http://whois.com).
- Enter the target domain (e.g., [example.com](http://example.com)).
- Record the information such as domain owner, registrar, creation date, expiry date, name servers, and contact emails.

### **2. Perform DNS Enumeration using nslookup/dig**

- Open Command Prompt / Terminal.
- Run the following commands:
- `nslookup example.com`
- `nslookup -type=MX example.com`
- `dig example.com ANY`
- Note the A, MX, NS, TXT, CNAME records.

### **3. Identify Emails & Metadata using Google Dorks**

- Open Google search.
- Use dorks such as:
- `site:example.com`
- `"@example.com"`
- `filetype:pdf site:example.com`
- Collect emails, documents, and metadata.



## Output:

### Step 1 : whois - Domain ownership info

```
root@kali: /home/kali
File Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# whois blinkit.com
Domain Name: BLINKIT.COM
Registry Domain ID: 2529784_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: http://www.godaddy.com
Updated Date: 2024-09-05T14:31:26Z
Creation Date: 1998-11-24T05:00:00Z
Registry Expiry Date: 2033-11-23T05:00:00Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: 480-624-2505
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: HENRY.NS.CLOUDFLARE.COM
Name Server: NINA.NS.CLOUDFLARE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-11-21T13:11:39Z <<<
```

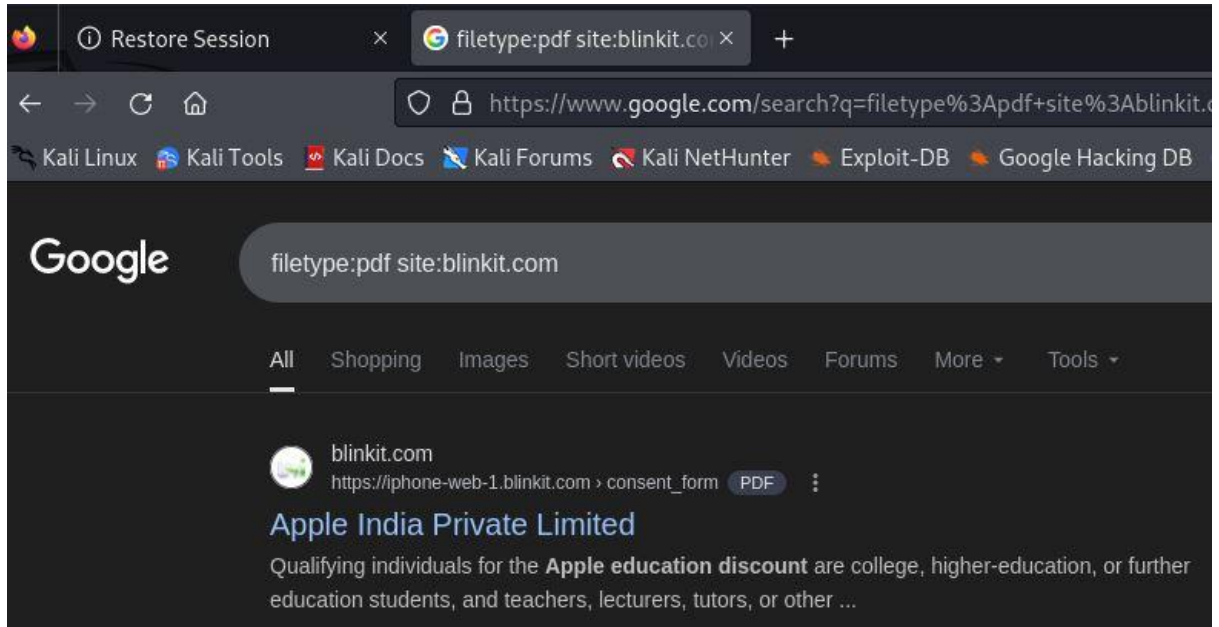
### Step 2 : nslookup - DNS record lookup

```
(root@kali)-[/home/kali]
# nslookup blinkit.com
Server: 192.168.198.2
Address: 192.168.198.2#53

Non-authoritative answer:
Name: blinkit.com
Address: 104.18.35.23
Name: blinkit.com
Address: 172.64.152.233
Name: blinkit.com
Address: 2606:4700:8d73:8ed:2c42:4ba:80c8:a651
```



### Step 3 : Google Dorking - Advanced search hacking.



### Step 4 : Report

footprinting report.txt - Notepad

File Edit Format View Help

#### Lab 1: Reconnaissance & Footprinting Report

Target Domain:blinkit.com

#### 1. WHOIS Lookup Results

The WHOIS lookup provides ownership and registration information about the target domain.

#### -Key Findings:

Domain Name: BLINKIT.COM  
Registrar: GoDaddy.com, LLC  
WHOIS Server: whois.godaddy.com  
Creation Date: 1998-11-24  
Updated Date: 2024-09-05  
Expiry Date: 2033-11-23  
Registrant Contact: Not directly visible (protected)  
Abuse Contact: [abuse@godaddy.com](mailto:abuse@godaddy.com)  
Name Servers:

- \* NINA.NS.CLOUDFLARE.COM
- \* CONDUSEC.NS.CLOUDFLARE.COM





## 2. DNS Lookup (nslookup) Results

DNS lookup returns the IP addresses and server information associated with the domain.

Output Summary:

Server Queried: 192.168.198.2

Resolved IP Addresses:

- \* 104.18.35.23
- \* 172.64.152.233
- \* 2606:4700:8d73:8ed:2c42:4ba:80c8:a651 (IPv6)

## 3. Google Dorking Results

Google Dorks were used to look for publicly accessible documents.

-Dork Used:

"filetype:pdf site:blinkit.com"

-Result:

A PDF belonging to Apple India Private Limited was discovered, hosted under a Blinkit subdomain.

## 4. Summary of Footprinting Findings

- \* Blinkit uses Cloudflare for DNS and security.
- \* WHOIS details are partially hidden for privacy.
- \* Multiple IP addresses indicate CDN usage.
- \* Google Dorks revealed accessible PDF documents.

---

-Conclusion

The reconnaissance exercise successfully gathered public information about the target domain without performing any intrusive actions. This type of passive footprinting helps understand the domain infrastructure and possible entry points for further analysis.



## **CONCLUSION :**

In this lab, we successfully performed reconnaissance and footprinting to gather essential information about a target domain. Using tools like Whois, nslookup, Google Dorks, and we collected domain details, DNS records, email IDs, metadata, and public information available on the internet. This process demonstrated how attackers and security professionals identify a target's online footprint. The lab helped us understand the importance of information gathering in the initial phase of ethical hacking and how it supports further security assessment and penetration testing.



**School of Computer Science, Engineering and Applications(SCSEA)**

**Academic Year: 2025-26**

**MCA S.Y. (Sem III)**

**Subject : Ethical Hacking & Digital Forensics (CS2001)**

**Name of the Student: Jayesh Bhushan Bhangale**

**PRN: 20240804069**

**Title of Practical : Network Scanning & Port Scanning using Nmap**

**AIM :** To study and perform network scanning and port scanning using Nmap in order to identify live hosts, open ports, running services, service versions, and the operating system of a target machine.

**THEORY :** Network scanning is the process of finding which devices are active in a network and what services they are running. Nmap is a tool used to check open ports, running services, and OS details of a target system.

Through scanning, we can:

- Identify live hosts
- Find open, closed, or blocked ports
- Detect which services (like HTTP, FTP, SSH) are running
- Know the version of each service
- Identify the operating system



## **PERFORM OPERATION :**

1. Start by discovering which hosts are active in the network.
2. Perform port scanning on the target to identify open, closed, and filtered ports.
3. Analyze the open ports to determine the types of services running on the target system.
4. Detect the version of each service to identify outdated or vulnerable software.
5. Perform OS fingerprinting to determine the operating system and system characteristics.
6. Evaluate the scanned information to identify potential vulnerabilities based on open ports and exposed services.
7. Prepare a report summarizing hosts found, ports discovered, service details, OS information, and possible security risks.



## OUTPUT:

1. **Step 1** : Scan a target machine using different Nmap scan types.

### nmap -sn - Host discovery scan

```
(root@kali)-[/home/kali]
# nmap -sn blinkit.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-23 01:39 EST
Nmap scan report for blinkit.com (172.64.152.233)
Host is up (0.0014s latency).
Other addresses for blinkit.com (not scanned): 104.18.35.23 2606:4700:8d73:8ed:2cfd:a69:80c8:a651
Nmap done: 1 IP address (1 host up) scanned in 0.76 seconds
```

### nmap - Network scanning tool

```
(root@kali)-[/home/kali]
# nmap blinkit.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-23 01:41 EST
Nmap scan report for blinkit.com (104.18.35.23)
Host is up (0.041s latency).
Other addresses for blinkit.com (not scanned): 172.64.152.233 2606:4700:8d73:8ed:2cd7:a6a:80c8:a651
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
8080/tcp   open  http-proxy
8443/tcp   open  https-alt
Nmap done: 1 IP address (1 host up) scanned in 87.90 seconds
```

### nmap -A - Aggressive scan details

```
(root@kali)-[/home/kali]
# nmap -A blinkit.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-23 02:35 EST
Nmap scan report for blinkit.com (104.18.35.23)
Host is up (0.022s latency).
Other addresses for blinkit.com (not scanned): 172.64.152.233 2606:4700:8d73:8ed:2cfe:a69:80c8:a651
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  tcpwrapped
|_http-server-header: cloudflare
443/tcp   open  tcpwrapped
|_ssl-cert: Subject: commonName=blinkit.com
| Subject Alternative Name: DNS:blinkit.com, DNS:*.blinkit.com
| Not valid before: 2025-09-29T12:57:17
|_Not valid after: 2025-12-28T13:56:42
|_http-server-header: cloudflare
8080/tcp   open  tcpwrapped
|_http-server-header: cloudflare
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host

TRACEROUTE (using port 80/tcp)
HOP RTT ADDRESS
1 ... 30

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.39 seconds
```



## Step 2 : Perform OS fingerprinting and service version detection.

### nmap -O - OS detection scan.

```
(root@kali)-[/home/kali]
# nmap -O blinkit.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-23 02:30 EST
Nmap scan report for blinkit.com (104.18.35.23)
Host is up (0.018s latency).
Other addresses for blinkit.com (not scanned): 172.64.152.233 2606:4700:9c63:8ed:2cd7:a6c:80c8:a651
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
8080/tcp  open  http-proxy
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.69 seconds
```

### nmap -sV: Service version scan

```
(root@kali)-[/home/kali]
# nmap -sV blinkit.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-23 02:33 EST
Nmap scan report for blinkit.com (172.64.152.233)
Host is up (0.020s latency).
Other addresses for blinkit.com (not scanned): 104.18.35.23 2606:4700:8d73:8ed:2cc6:a6b:80c8:a651
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE  VERSION
80/tcp    open  tcpwrapped
443/tcp   open  tcpwrapped
8080/tcp  open  tcpwrapped
8443/tcp  open  tcpwrapped
```

## Step 3 - Identify potential vulnerabilities based on open ports

### nmappotential - Find open-port risks

```
----Potential Vulnerabilities Based on Open Ports-----
Target: blinkit.com

Open Ports Detected:

*80/tcp - HTTP
Port 80 (HTTP): Unencrypted traffic may expose data and allow attackers to intercept or modify requests.

*443/tcp - HTTPS
Port 443 (HTTPS): Possible weak SSL/TLS settings or certificate issues that could be exploited.

*8080/tcp - http-proxy
Port 8080 (HTTP-Proxy): Often used for admin panels or proxy services; may expose sensitive interfaces if misconfigured.

*8443/tcp - https-alt
Port 8443 (HTTPS-Alt): Alternate secure web service; could host outdated or unprotected applications.
```



**Conclusion :** In this lab, we successfully used Nmap to scan a target system, identify open ports, detect running services, and perform OS fingerprinting. By analyzing these results, we learned how open ports and exposed services can reveal potential security vulnerabilities. This lab helps in understanding how attackers and security professionals assess a network's security posture using effective scanning techniques.



**School of Computer Science, Engineering and Applications(SCSEA)**

**Academic Year: 2025-26**

**MCA S.Y. (Sem III)**

**Subject : Ethical Hacking & Digital Forensics (CS2001)**

**Name of the Student: Jayesh Bhushan Bhangale**

**PRN: 20240804069**

**Title of Practical : Vulnerability Scanning using Nessus / OpenVAS**

**AIM :** To understand the vulnerability assessment process and perform vulnerability scanning on a target system using Nessus or OpenVAS.

**THEORY :**

Vulnerability scanning is the process of automatically detecting security weaknesses in systems, networks, and applications. Tools like Nessus and OpenVAS are widely used by security professionals to identify known vulnerabilities, misconfigurations, outdated software, and weak security settings.

**Key Concepts:**

- **Vulnerability:**  
A flaw or weakness that can be exploited by attackers.
- **Vulnerability Scanner:**  
A tool that checks systems against known vulnerability databases (CVEs).
- **Scan Policies:**  
Predefined or custom settings that decide how deep or detailed the scan will be (basic, advanced, credentialed, etc.).
- **Scan Report:**  
A summary showing detected vulnerabilities with their severity levels (Low, Medium, High, Critical).

Vulnerability scanning helps organizations identify and fix security issues before attackers misuse them.



## **PERFORM OPERATION:**

### **Configure Scan Policy**

- Choose a basic or advanced scan type based on the target.
- Set scanning options like port range, service detection, and vulnerability checks.

### **Run Vulnerability Scan**

- Start the scan on the target system.
- Wait for the tool to analyze open ports, services, and known weaknesses.

### **Analyze Scan Results**

- Review detected vulnerabilities with their severity levels.
- Understand risk ratings such as Low, Medium, High, or Critical.

### **Export Scan Report**

- Save the generated report in PDF/HTML/CSV format.
- Use the report to plan remediation steps.



## OUTPUT:

### STEP 1 - Update System - `sudo apt update`

```
(kali㉿kali)-[~]  
$ sudo apt update  
[sudo] password for kali:  
Hit:1 http://http.kali.org/kali kali-rolling InRelease  
1144 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

### STEP 2 — Install OpenVAS (Greenbone) - `sudo apt install openvas -y`

```
(kali㉿kali)-[~]  
$ sudo apt install gvm -y  
Installing:  
gvm  
  
Installing dependencies:  
greenbone-security-assistant  gvm-tools  python3-terminaltables3  
gsad  libmicrohttpd12t64  
  
Suggested packages:  
python3-terminaltables3-doc  
  
Summary:  
Upgrading: 0, Installing: 6, Removing: 0, Not Upgrading: 1144  
Download size: 3,927 kB  
Space needed: 16.8 MB / 8,357 MB available  
  
Get:1 http://kali.download/kali kali-rolling/non-free amd64 greenbone-security-assistant all 25.3.1-0kali1 [3,455 kB]  
Get:2 http://kali.download/kali kali-rolling/main amd64 libmicrohttpd12t64 amd64 1.0.2-2 [156 kB]  
Get:3 http://kali.download/kali kali-rolling/main amd64 gsad amd64 24.7.0-1 [128 kB]  
Get:5 http://kali.download/kali kali-rolling/main amd64 python3-terminaltables3 all 4.0.0-4 [15.3 kB]  
Get:6 http://kali.download/kali kali-rolling/main amd64 gvm-tools all 25.4.1-1 [161 kB]  
Get:4 http://mirrors.esto.network/kali kali-rolling/main amd64 gvm all 25.04.1 [11.8 kB]  
Fetched 3,927 kB in 1s (3,071 kB/s)  
Selecting previously unselected package greenbone-security-assistant.  
(Reading database ... 416653 files and directories currently installed.)  
Preparing to unpack .../0-greenbone-security-assistant_25.3.1-0kali1_all.deb ...  
Unpacking greenbone-security-assistant (25.3.1-0kali1) ...  
Selecting previously unselected package libmicrohttpd12t64:amd64.  
Preparing to unpack .../1-libmicrohttpd12t64_1.0.2-2_amd64.deb ...  
Unpacking libmicrohttpd12t64:amd64 (1.0.2-2) ...  
Selecting previously unselected package gsad.  
Preparing to unpack .../2-gsad_24.7.0-1_amd64.deb ...  
Unpacking gsad (24.7.0-1) ...  
Selecting previously unselected package gvm.  
Preparing to unpack .../3-gvm_25.04.1_all.deb ...  
Unpacking gvm (25.04.1) ...  
Selecting previously unselected package python3-terminaltables3.  
Preparing to unpack .../4-python3-terminaltables3_4.0.0-4_all.deb ...  
Unpacking python3-terminaltables3 (4.0.0-4) ...
```

### STEP 3 — Setup OpenVAS Environment - `sudo gvm-setup`

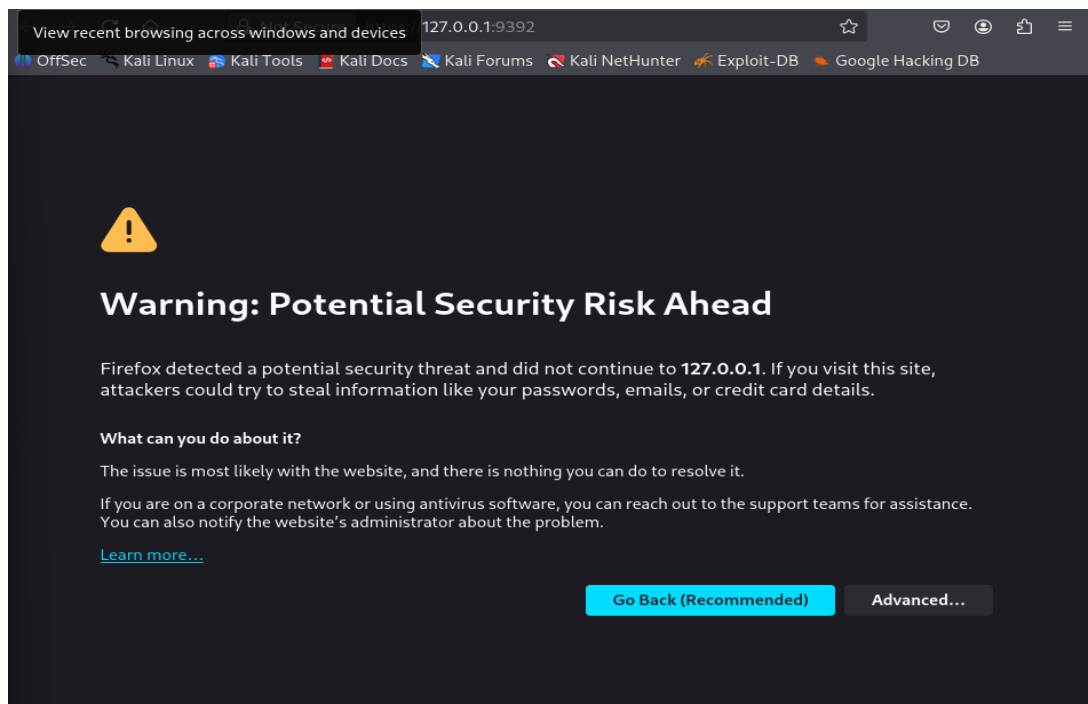
Note username and password

```
[+] Done  
[*] Please note the password for the admin user  
[*] User created with password '097b8156-604c-4d54-b284-0dfb999637aa'.  
  
[>] You can now run gvm-check-setup to make sure everything is correctly configured  
  
(kali㉿kali)-[~]  
$
```

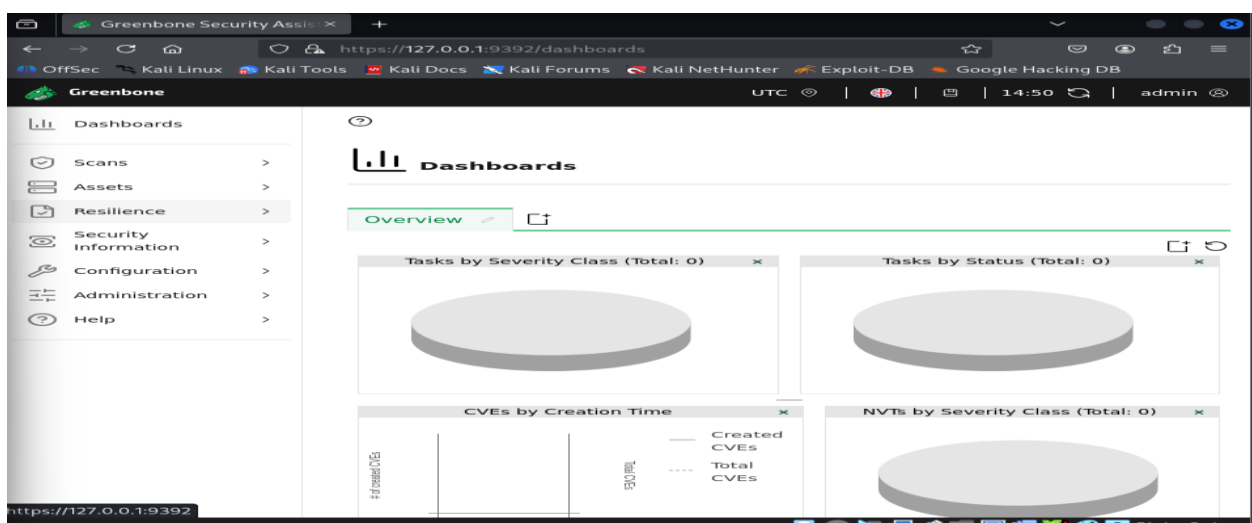


**STEP 4 —** Start All GVM Services - `sudo gvm-start`

**STEP 5 —** Open the Web Interface



**STEP 6 —** Login





## **STEP 7 —** Create a Target for Scanning

1. Go to **Configuration → Targets**
2. Click **New Target**
3. Enter:
  - **Name:** Test Scan
  - **Hosts:** 10.0.2.15

Greenbone Security Assistant

What Is My IP Address - S

https://127.0.0.1:9392/targets

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Greenbone UTC 14:36 admin

Dashboards

Scans >

Assets >

Resilience >

Security Information >

Configuration >

**Targets**

Port Lists

Credentials

Scan Configs

Alerts

Schedules

Report Configs

Report Formats

Filter

Targets 1 of 1

1 - 1 of 1

Name ↑	Hosts ↑↓	IPs ↑↓	Port List ↑↓	Credentials	Actions
test scan	10.0.2.15	1	All IANA assigned TCP		

Apply to page contents

(Applied filter: sort=name first=1 rows=10)

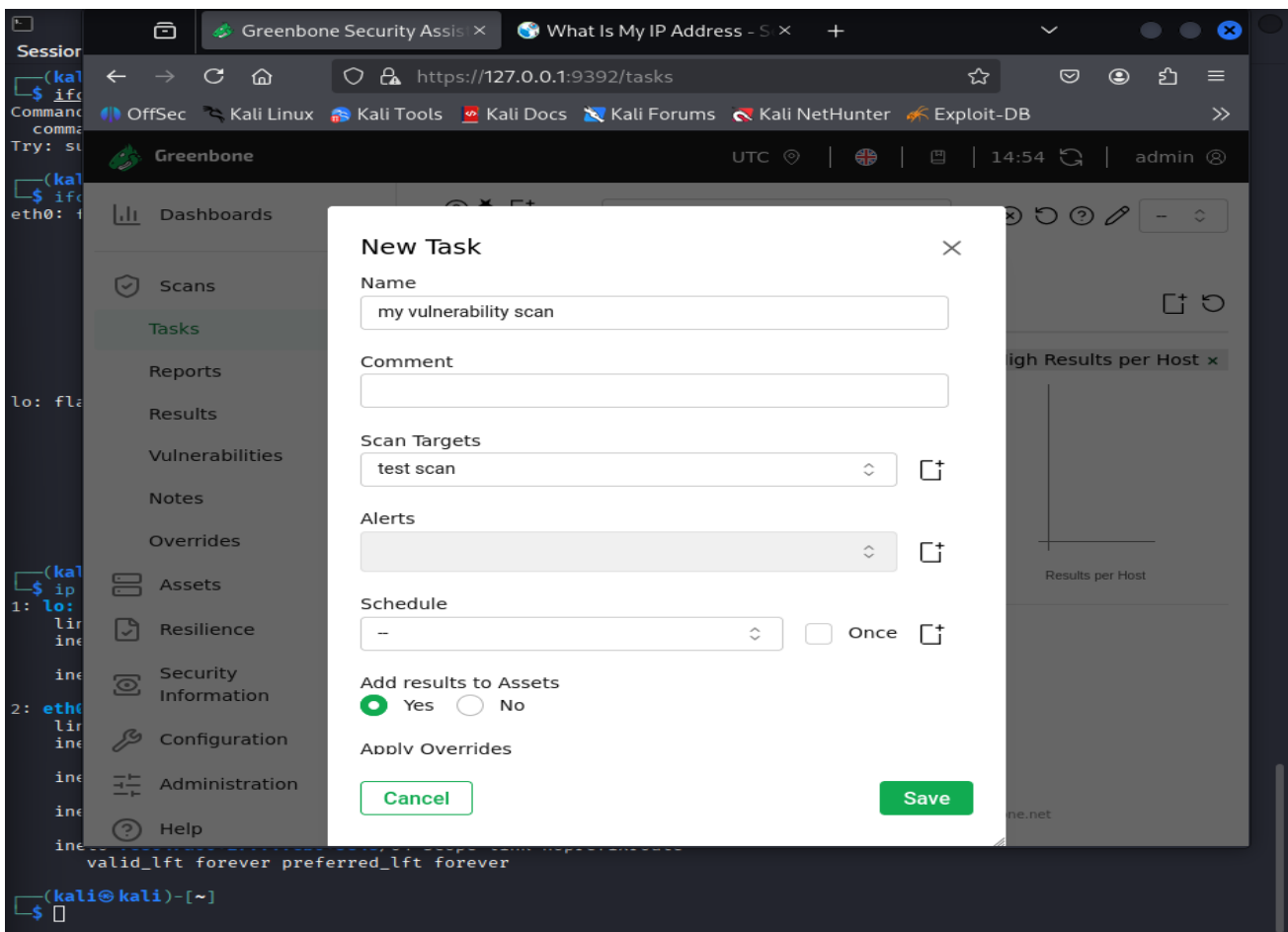
1 - 1 of 1

Copyright © 2009-2025 by Greenbone AG, www.greenbone.net



## **STEP 8 —** Create a Scan Task

1. Go to **Scans** → **Tasks**
2. Click **New Task**
3. Fill:
  - **Name:** My Vulnerability Scan
  - **Target:** test scan
  - **Scanner:** *OpenVAS Default*





**Step 9:** Run the Scan

- Select the scan task → Click **Start**.
- Monitor progress. Depending on the number of hosts and network size, it may take some time.

**Step 10 :** Review Scan Results

1. After completion, go to “**Reports**” → **Select your task**.
2. Review:
  - High, Medium, Low severity vulnerabilities
  - CVE references (Common Vulnerabilities and Exposures)
  - Suggested mitigations

**CONCLUSION :** In this lab, we successfully performed vulnerability scanning using Nessus / OpenVAS to identify security weaknesses in a target system. We learned how to configure scan policies, run scans, and analyze the results. The generated report helped us understand the severity of detected vulnerabilities and their potential impact. This lab provided a practical understanding of how vulnerability assessment works and why it is an important step in strengthening the security of networks and systems.



**School of Computer Science, Engineering and Applications(SCSEA)**

**Academic Year: 2025-26**

**MCA S.Y. (Sem III)**

**Subject : Ethical Hacking & Digital Forensics (CS2001)**

**Name of the Student: Jayesh Bhushan Bhangale**

**PRN: 20240804069**

**Title of Practical : Password Cracking Basics**

**AIM :** To understand the basic concepts of password cracking and perform simple brute-force and dictionary attacks using tools like John the Ripper or Hydra.

**THEORY :**

Password cracking is the process of recovering passwords from stored data. It is used by penetration testers to check the strength of user passwords.

Common techniques include:

- **Dictionary Attack:**  
Uses a list of common or known passwords. Fast but limited.
- **Brute Force Attack:**  
Tries every possible combination until the correct password is found.  
Accurate but time-consuming.

Password cracking helps identify weak, guessable, or reused passwords and improves overall system security.



### **PERFORM OPERATION:**

1. Load a sample password file or hashed password into the tool.
2. Select the attack type (dictionary or brute force).
3. Run the cracking process and wait for the tool to test passwords.
4. Observe the cracked password and record the result.
5. Analyze why the password was cracked easily and note how password strength can be improved.



## OUTPUT:

**Step 1 -** Load sample password file.

**nano** - Simple text editor

```
(root@kali)-[/home/kali/Downloads]
# nano sample.txt
```

**cat** - View file content

```
(root@kali)-[/home/kali/Downloads]
# cat sample.txt
user1:$1$zPzafods$YH./1mtA9R7HWR0b0Twit0:1000:1000::/home/user1:/bin/bash
test:$1$w74blgv2$5AZOACZLP0s1eW0ekfjG51:1000:1000::/home/test:/bin/bash
demo:$1$EkLgbaxJ$jYd0TPl9BHsn07XdfgPDc.:1000:1000::/home/demo:/bin/bash
```

**Step 2 -** Run brute force/dictionary attack.

**john**: Password-cracking tool

```
(root@kali)-[/home/kali/Downloads]
# john sample.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4x3])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 18 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password      (test)
123            (user1)
admin          (demo)
3g 0:00:00:01 DONE 2/3 (2025-11-23 07:05) 2.678g/s 4875p/s 5277c/s 5277C/s nina..buzz
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```



**Step 3** - Note cracked password.

**john --show**: Display cracked passwords

```
(root@kali)-[/home/kali/Downloads]
# john --show sample.txt
user1:123:1000:1000::/home/user1:/bin/bash
test:password:1000:1000::/home/test:/bin/bash
demo:admin:1000:1000::/home/demo:/bin/bash

3 password hashes cracked, 0 left
```

**CONCLUSION :** In this lab, we learned how basic password-cracking tools like John the Ripper and Hydra work. By running brute-force and dictionary attacks on a sample password file, we observed how weak passwords can be easily cracked. This activity helps in understanding the importance of strong, complex passwords and how attackers exploit weak ones, reinforcing good password security practices.

