Comprehensive Report
by
Shaun Franklyn

ABSTRACT:

In today’s interconnected digital environment, information systems and web applications are increasingly exposed to security threats, making proactive security assessment a critical requirement for organizations. This report focuses on the theoretical understanding and practical implementation of Vulnerability Assessment and Penetration Testing (VAPT) using open-source and freely available tools, without reliance on commercial security solutions. The objective of this study is to evaluate systems for potential security weaknesses, simulate real-world attack scenarios, and recommend effective remediation strategies in alignment with industry best practices.

The assessment follows a structured VAPT methodology consisting of planning, discovery, exploitation, risk analysis, and reporting phases. Security testing techniques such as vulnerability scanning, penetration testing, and compliance validation are performed using tools including Nmap, OpenVAS, Nikto, OWASP ZAP, and the Metasploit Framework. A controlled virtual environment is created using Kali Linux and intentionally vulnerable machines such as Metasploitable, enabling safe and ethical testing.

Identified vulnerabilities are analyzed and prioritized using the Common Vulnerability Scoring System (CVSS) and a qualitative risk matrix based on likelihood and impact. The study also emphasizes alignment with widely accepted security standards and frameworks, including NIST guidelines, OWASP Top 10, CIS Benchmarks, ISO/IEC 27001, GDPR, and HIPAA, to ensure comprehensive coverage of security and compliance requirements.

All findings are systematically documented using standard reporting practices, incorporating technical evidence such as screenshots, affected services, CVE references, and CVSS scores. The final report presents an executive summary for management, detailed technical findings for security teams, and clear remediation recommendations including patching, configuration hardening, and security best practices. This study demonstrates how effective security assessments can be conducted using open-source tools while building strong foundational skills in cybersecurity assessment, risk management, and professional security documentation.

OBJECTIVE:

The primary objective of this task is to gain theoretical and practical knowledge of security assessment and VAPT methodologies using free and open-source tools. Specific objectives include:

- To understand the fundamentals of security assessment and different types of security testing
- To evaluate systems for vulnerabilities using open-source tools instead of paid solutions
- To apply a structured VAPT methodology covering planning, discovery, attack, and reporting phases
- To identify common network and web application vulnerabilities through practical lab exercises
- To assess and prioritize security risks using CVSS scoring and risk matrices
- To develop effective documentation and reporting skills aligned with industry standards

INTRODUCTION:

With the rapid growth of digital systems and web-based applications, organizations face increasing risks from cyber threats and security vulnerabilities. Regular security assessments are essential to identify weaknesses before they can be exploited by malicious actors. Vulnerability Assessment and Penetration Testing (VAPT) plays a critical role in strengthening system security by systematically discovering, validating, and prioritizing vulnerabilities.

This task focuses on learning how to perform security assessments using open-source tools and standard frameworks, making the process accessible without the need for expensive commercial software. The study follows established methodologies such as the OWASP Web Security Testing Framework and aligns findings with recognized standards including ISO 27001, GDPR, and OWASP Top 10.

A controlled virtual environment using Kali Linux and vulnerable machines such as Metasploitable is utilized to simulate real-world attack scenarios. Vulnerabilities are identified, assessed for risk, and documented in a structured report that includes executive summaries, technical findings, and remediation recommendations. This approach provides a strong foundation for understanding practical cybersecurity assessment techniques and professional reporting practices.

Theoretical Knowledge

1. Understanding Security Assessment

- **Security Assessment:** This is the broad, overarching process. Using a framework like the NIST Cybersecurity Framework provides a strategic structure. For assessment, the "Identify" and "Detect" pillars are most relevant. You are identifying assets and detecting vulnerabilities.

- **Vulnerability Assessment:** This is a systematic review of security weaknesses. OpenVAS (now part of the Greenbone Vulnerability Management) is the premier open-source scanner. It checks systems against a massive database of known vulnerabilities (CVEs) and provides a detailed report, often including CVSS scores. It answers the question, "What are the known weaknesses on my system?"

- **Penetration Testing:** This is an authorized, simulated attack. It goes beyond finding vulnerabilities to actively exploiting them to determine the real-world impact. Kali Linux is the industry-standard platform, pre-loaded with tools like Metasploit for exploitation and Nmap for discovery. It answers the question, "Can an attacker actually break in and what can they do?"

- **Compliance Testing:** This is about checking boxes against a standard. CIS Benchmarks are hardening guides for specific operating systems and software. You use a checklist to verify configurations (e.g., "Is password complexity enforced?"). It ensures you are following best practices but doesn't guarantee the absence of vulnerabilities.

2. VAPT Methodology

A structured methodology is non-negotiable for a professional and thorough assessment.

- **Planning:** The most critical phase. Define the Rules of Engagement (RoE) in writing. What are the target IP ranges and domains? What time windows are allowed? What types of attacks are explicitly forbidden (e.g., Denial of Service)? Dradis CE is purpose-built for this, acting as a central repository for scope, notes, and findings.
- **Discovery (Reconnaissance):** Information gathering is the foundation.
  - Network: Use Nmap to map the attack surface. A command like nmap -sV -O -p- target_ip will identify open ports, the services running, their versions, and even the host OS.
  - Web: Use OWASP ZAP as a proxy in your browser. As you crawl the target application, ZAP passively maps the site structure, identifies forms and parameters, and spiders for content, building a complete picture of the application's attack surface.
- **Attack (Exploitation):** This is the active phase. If Nmap finds an outdated service, you search Metasploit (search [service_name] [version]) for an exploit. You load the module, set the required options (like RHOSTS for the target IP), and execute it to gain access (e.g., a reverse shell).
- **Reporting:** The final and most important deliverable. The report must be clear, concise, and actionable. It should explain the business impact of each finding and provide clear, step-by-step instructions for remediation.

3. Security Standards & Compliance

Understanding the "why" behind security is crucial.

- **GDPR/HIPAA:** These are regulations focused on data privacy. A data breach caused by an unpatched vulnerability (e.g., SQL Injection) can lead to severe financial penalties and reputational damage. Compliance is a form of risk management.

- **ISO 27001:** This is a management system standard. It's about creating a comprehensive, repeatable process for managing information security within an organization, covering people, processes, and technology.

- **OWASP Top 10:** This is your practical, priority-based guide for web application security. It lists the 10 most critical security risks (e.g., Injection, Broken Authentication, XSS). If you can find and fix these, you have mitigated the most common and dangerous web threats.

4. Risk Assessment Basics

We cannot fix everything at once. Risk assessment helps you prioritize your efforts.

- **CVSS (Common Vulnerability Scoring System):** This is the universal standard for rating the severity of a vulnerability. It's a 0-10 score based on factors like Attack Vector, Complexity, and Impact. The NVD CVSS Calculator is the definitive tool. A CVSS score of 9.8 is critical and easily exploitable, while a 3.1 is a lower priority.
- **Risk Matrix:** CVSS measures severity, but risk is Likelihood x Impact. A critical vulnerability on an isolated, internal server is lower risk than a medium vulnerability on a public-facing login page. A 3x3 or 5x5 matrix (Likelihood vs. Impact) is a simple yet powerful way to visualize this and decide what to fix first.

5. Common Vulnerabilities

To identify and understand frequently occurring security vulnerabilities through hands-on practice using controlled lab environments and security tools.

Network Vulnerabilities

These vulnerabilities arise from poor system or network configurations, including:

- Unnecessary open ports
- Misconfigured services
- Such issues can be identified and analyzed using tools like Nmap.

Web Application Vulnerabilities

Common web-based security flaws include:

- SQL Injection (SQLi) – Exploiting insecure database queries
- Cross-Site Scripting (XSS) – Injecting malicious scripts into web pages

These vulnerabilities can be safely practiced and tested using intentionally vulnerable applications such as OWASP Juice Shop.

How to Learn

Gain practical experience by working with deliberately vulnerable platforms, including:

- Metasploitable – A vulnerable virtual machine designed for security testing and exploitation practice
- VulnHub – A collection of vulnerable virtual machines for hands-on penetration testing and learning

6. Documentation Fundamentals in Security Assessments

Objective: Master the creation of professional, structured reports using free and open-source tools to document findings effectively.

Explanation:

Professional documentation is a critical phase in Vulnerability Assessment and Penetration Testing (VAPT). A well-structured report communicates findings clearly to technical and non-technical stakeholders, includes evidence (screenshots, logs), risk prioritization, and actionable remediation steps. Good reporting tools help organize notes during testing, import tool outputs (e.g., from Nmap, OpenVAS, Nikto), and generate polished final reports (PDF/Word/HTML).

Dradis Community Edition (CE)

- **Description:** Open-source collaboration and reporting platform specifically designed for InfoSec teams. It centralizes findings, supports team collaboration, imports results from 30+ tools (Nmap, Nessus/OpenVAS, Burp, Metasploit, etc.), and exports to Word/PDF/HTML with customizable templates.
- **Why it's great for pentesting**: Automates merging of scan results, tracks issues, methodologies, and evidence in one place.
- **Installation:** Available on GitHub. Easy Docker deployment for quick setup (e.g., community Docker images like dafal/dradis-docker-ce).
- **Best for:** Full-featured reporting workflows.

CherryTree

- **Description:** Hierarchical note-taking application with rich text, syntax highlighting, image embedding, tables, and code boxes. Stores data in a single encrypted file (SQLite or XML).
- **Why it's great for pentesting:** Excellent for organizing raw notes during testing – create nodes for Recon, Vulnerabilities, Exploits, Screenshots, Commands, etc. Supports password protection and easy export to PDF/HTML.
- **Installation:** Available on GitHub. Pre-installed in Kali Linux.
- **Bonus:** There are pentest-specific templates, e.g., OSCP-style CherryTree template on GitHub (search for "CherryTree-OSCP-Template") with pre-built structures for phases like reconnaissance, exploitation, and reporting.
- **Best for:** Personal/technical note-taking during active testing.

Other Standard Free/Open-Source Reporting Tools

- **PwnDoc**: Modern pentest report generator. Web-based, customizable findings library, exports to Docx/PDF. Great for structured vulnerability descriptions.
- **WriteHat**: Markdown-based reporting tool that converts to HTML/PDF. Drag-and-drop components, finding databases – frees you from Word.
- **SysReptor**: Open-source alternative with customizable templates (mentioned as a strong Dradis-like option).
- **LaTeX Templates**: For beautiful PDFs, use repositories like robingoth/pentest-report-template or profi248/pentest-report (includes graphical CVSS scoring).
- **Simple Options**: LibreOffice/Markdown with Pandoc for conversion to PDF, or Google Docs/Excel for basic tracking.

How to Learn:
Using Free Templates from GitHub
The best way to get started quickly is by downloading and customizing free templates:

- Search GitHub for "pentest report template" or "vulnerability assessment report template".

Popular free repositories:

- https://github.com/hmaverickadams/TCM-Security-Sample-Pentest-Report (Simple Word-based starter template with one finding example).
- https://github.com/reconmap/pentest-reports (Collection of real-world templates from top security companies).
- https://github.com/MTK911/pentest-report-template (Basic application testing template).
- https://github.com/pwndoc/pwndoc (Built-in customizable templates).
- LaTeX options for professional layouts: https://github.com/robingoth/pentest-report-template or https://github.com/profi248/pentest-report.

Practical Application

Setup Testing Environment
Objective: Create an Isolated Virtual Lab

- The goal is to have a Kali Linux "attacker" machine and a Metasploitable "victim" machine running in a completely isolated network. This prevents any accidental damage to your host computer or your local network.

Step 1: Install Virtualization Software
You need a virtualization platform to run virtual machines (VMs). The best free option is VirtualBox.

- Download: Go to the official VirtualBox website: https://www.virtualbox.org/wiki/Downloads
- Select: Download the version for your host operating system (Windows, macOS, or Linux).
- Install: Run the installer and accept all default settings. You may be prompted to install the "VirtualBox Extension Pack" during the process or as a separate download; this is recommended for enhanced functionality like USB support.

Step 2: Download and Install Kali Linux
Kali will be your primary attack machine, loaded with all the necessary tools.

1. Download: Navigate to the Kali Linux downloads page: https://www.kali.org/get-kali/
2. Select: Choose the "VirtualBox" image. This is a pre-packaged virtual machine that is much easier to set up than installing Kali from scratch.
3. Import:
    - Open VirtualBox.
    - Go to File -> Import Appliance....
    - Navigate to where you downloaded the Kali .ova file and select it.
    - Click "Next". You will be shown the VM's settings. You can adjust the amount of RAM or CPU cores if you wish, but the defaults are usually fine.
    - Click "Import". This will take a few minutes.

- Start Kali: Once imported, select the Kali VM in the VirtualBox manager and click the green "Start" arrow.
- Login: The default credentials for Kali are:
  - Username: kali
  - Password: kali

NOTE: I already had it installed on my PC

- **Step 3: Download and Import Metasploitable 3**
- Important Note: Metasploitable 3 is more complex to set up than Metasploitable 2 because it is built for modern Windows and Linux vulnerabilities and requires a more involved build process. For beginners,

- Here are the instructions
- **Metasploitable 3**
  - Metasploitable 3 is not available as a pre-built download. You must build it yourself using Vagrant and Packer.
- **Prerequisites:** You must have **Vagrant** and **Packer** installed on your host system (not inside a VM).
- **Download:** Clone the official GitHub repository:
  - bash
  - git clone https://github.com/rapid7/metasploitable3.git
  - cd metasploitable3
- **Build:** Follow the build instructions in the repository's README.md file. The command is typically vagrant up, which will automatically download the necessary base OS files (e.g., Windows Server 2008) and build the vulnerable VMs. **This process can take several hours and requires significant disk space and RAM.**

- **Step 4: Configure the Isolated Network**
- This is the most important configuration step. You need to place your VMs on a private, host-only network so they can communicate with each other but not with your main network or the internet.

Create a Host-Only Network:

- In VirtualBox, go to File -> Host Network Manager.
- Click the "Create" button (looks like a plus sign) to create a new network adapter. A new adapter (e.g., vboxnet0) will appear.
- Note the IP address assigned to this adapter (e.g., 192.168.56.1). This is your host machine's IP on the virtual network. Ensure the "DHCP Server" tab is enabled, as this will automatically assign IPs to your VMs.

Configure the Kali Linux VM:

- Shut down the Kali VM if it's running.
- In VirtualBox, select the Kali VM and go to Settings -> Network.
- Select Adapter 1.
- Set the "Attached to:" dropdown to Host-only Adapter.
- Set the "Name:" dropdown to the host-only adapter you just created (e.g., vboxnet0).
- Click "OK".

Configure the Metasploitable VM:

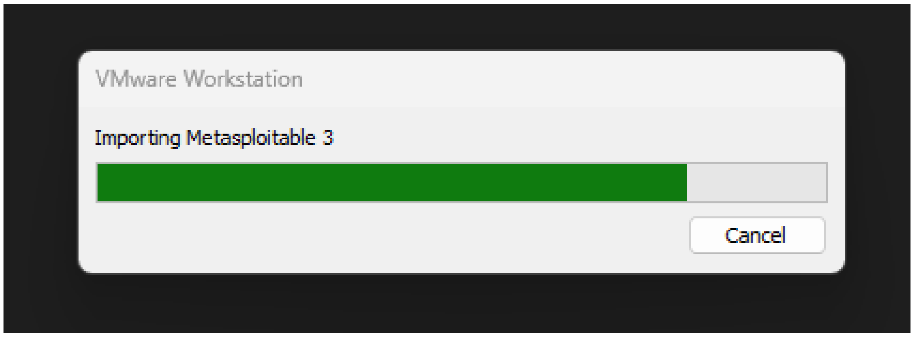
- Shut down the Metasploitable VM.
- In VirtualBox, select the Metasploitable VM and go to Settings -> Network.
- Select Adapter 1.
- Set the "Attached to:" dropdown to Host-only Adapter.
- Set the "Name:" dropdown to the same host-only adapter (e.g., vboxnet0).
- Click "OK".

Step 5: Verify the Lab Setup

- Start both the Kali and Metasploitable VMs.
- Once Kali is booted, open a Terminal.
- Find Kali's IP address on the private network by running:
- bash

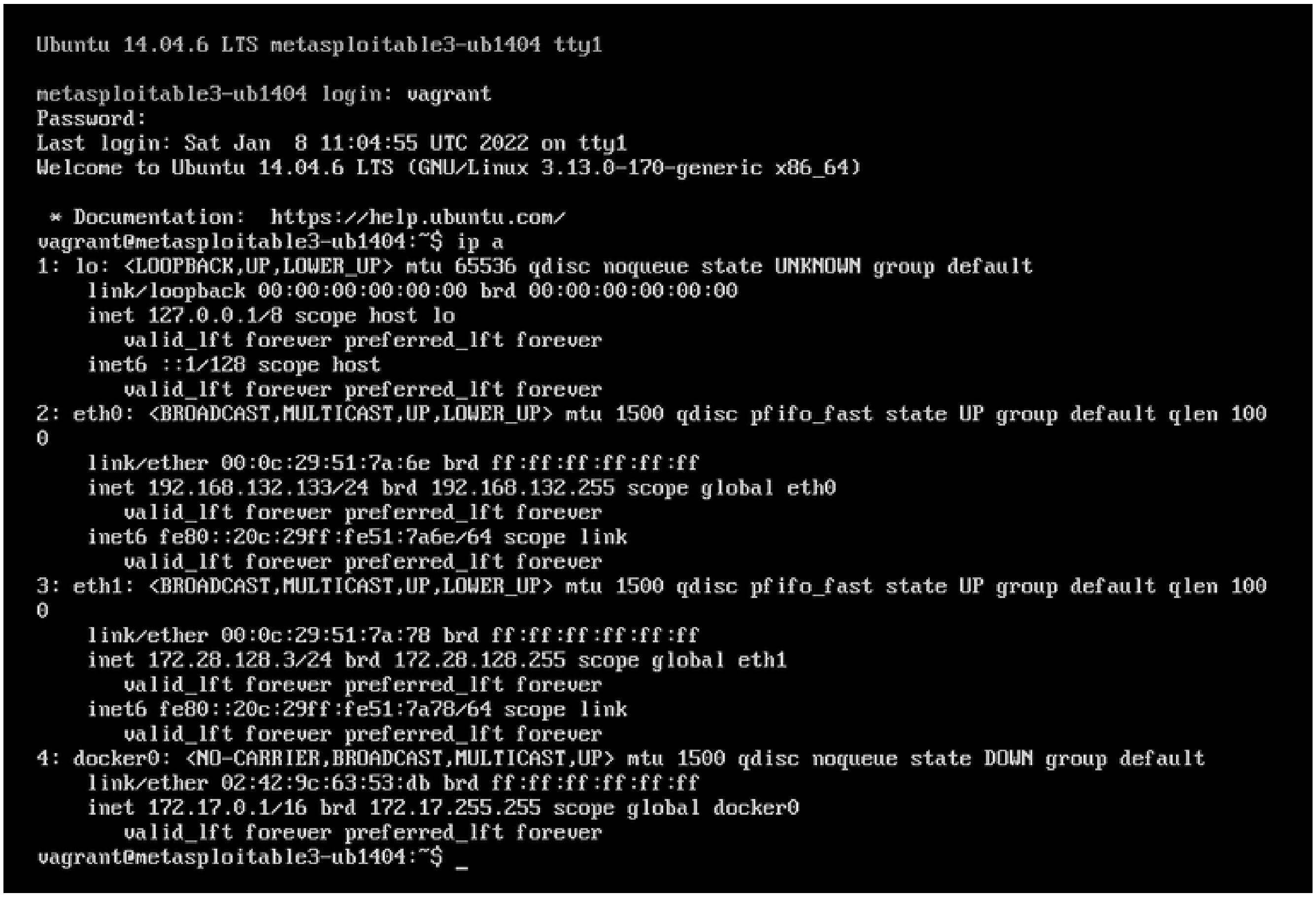
1. ip a
   (Look for the interface (e.g., eth0) that has an IP address in the 192.168.56.x range.)

- Scan the private network to find the Metasploitable VM's IP address using Nmap:



VMware Workstation

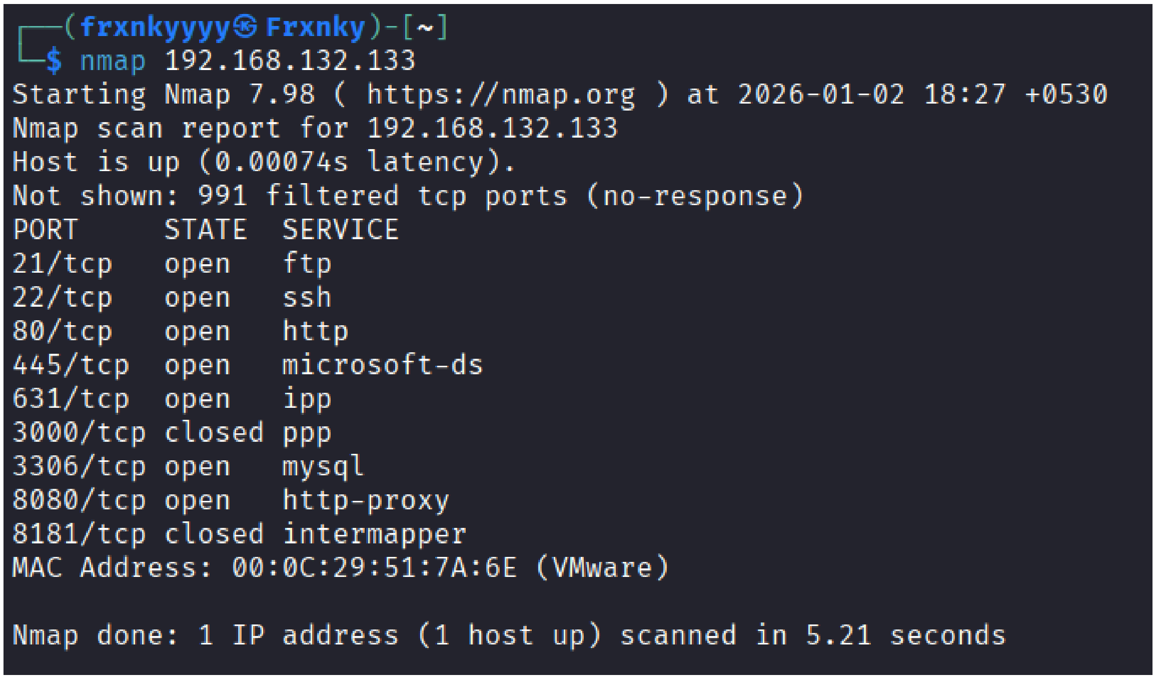
Importing Metasploitable 3



```
Ubuntu 14.04.6 LTS metasploitable3-ub1404 tty1

metasploitable3-ub1404 login: vagrant
Password:
Last login: Sat Jan  8 11:04:55 UTC 2022 on tty1
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
vagrant@metasploitable3-ub1404:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 100
0
    link/ether 00:0c:29:51:7a:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.132.133/24 brd 192.168.132.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe51:7a6e/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 100
0
    link/ether 00:0c:29:51:7a:78 brd ff:ff:ff:ff:ff:ff
    inet 172.28.128.3/24 brd 172.28.128.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe51:7a78/64 scope link
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:9c:63:53:db brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
vagrant@metasploitable3-ub1404:~$ _
```



```
┌──(frxnkyyyy㉿Frxnky)-[~]
└─$ nmap 192.168.132.133
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-02 18:27 +0530
Nmap scan report for 192.168.132.133
Host is up (0.00074s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper
MAC Address: 00:0C:29:51:7A:6E (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.21 seconds
```

2. Vulnerability Scanning

Objective: Identify Vulnerabilities on Metasploitable using OpenVAS

We will use the Greenbone Vulnerability Management (GVM) suite, which is the modern, open-source successor to OpenVAS. It is integrated into Kali Linux.

Step 1: Setup and Launch OpenVAS (GVM)

The first time you run GVM, you need to perform a one-time setup to create the user, database, and certificates. This process can take 15-30 minutes.

Open a Terminal in your Kali Linux VM.

Run the Setup Script: This script configures everything for you. It will download the latest vulnerability feeds and set up an admin user.
bash

- sudo gvm-setup
  - When prompted, create a password for your admin user. Make it strong and remember it.

Start the Services: Once the setup is complete, start the GVM services.
bash

- sudo gvm-start

This will launch the web server and all necessary background scanners. You can check their status with sudo gvm-check-setup.

Access the Web Interface:

- Open the Firefox web browser in your Kali VM.
- Navigate to: https://127.0.0.1:9392
- You will see a security warning because it's using a self-signed certificate. Click "Advanced" and then "Proceed to 127.0.0.1 (unsafe)".
- Log in with the username admin and the password you created during the setup.

Step 2: Configure the Scan

A. Define the Target

This tells GVM what system to scan.

1. From the top menu, navigate to **Configuration -> Targets**.
2. Click the **New Target** icon (a star with a plus sign).
3. Fill in the form:
   - **Name:** Metasploitable 3 Lab VM
   - **Hosts:** [IP_Address_of_Metasploitable] (192.168.132.133 In my case)
   - Leave the other settings as default for now. The "Alive Test" defaults to a combination of ICMP, TCP-ACK, and ARP ping, which is perfect for a local lab.
4. Click **Create**.

B. Define the Scan Task

This tells GVM how to scan the target.

- From the top menu, navigate to Scans -> Tasks.
- Click the New Task icon (a magic wand with a plus sign).
- Fill in the form:
  - Name: Full and Fast Scan of Metasploitable 3
  - Scan Target: Click the "wand" icon and select the Metasploitable 3 Lab VM target you just created.
  - Scanner: Keep the default OpenVAS Scanner.
  - Scan Config: Click the "wand" icon and select Full and fast. This is the most common scan profile, checking for thousands of vulnerabilities in a reasonable amount of time.
- Click Create.

Step 3: Run the Scan and Monitor Progress

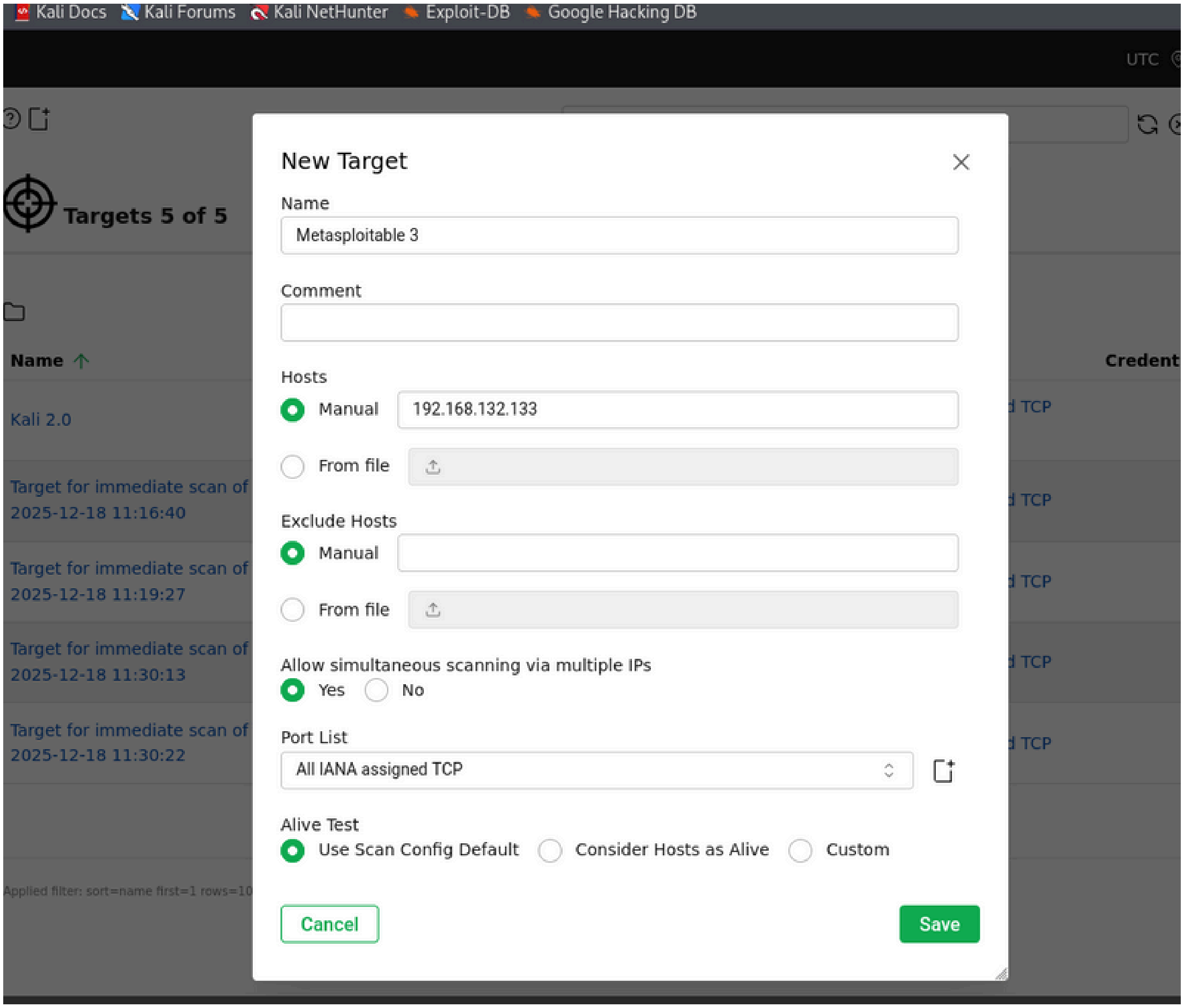
Now that the task is configured, we can start it.

1. On the Tasks page, you will see your new scan task. Its status will be "Scheduled".
2. Select the task by clicking the checkbox next to its name.
3. Click the **Start** button (a play icon) at the top of the task list.
4. The status will change to "Requested", then "Running", and finally "Done".

Step 4: Analyze the Results

This is where you turn the raw data into actionable intelligence.

1. Access the Report: Once the scan status is "Done", click on the task name. In the "Results" section, click on the date/time of the completed scan to view the full report.
2. Filter by Severity: The report screen is overwhelming at first. The first thing to do is filter by severity. On the left-hand side, you will see filters for High, Medium, and Low. Click on High to see the most critical findings first.
3. Examine a Critical Finding:
   - Look for a vulnerability like "DistCC Daemon Command Execution".
   - Click on the vulnerability name. You will see a detailed view.
   - Key Information to Note:
     - CVE ID: This is the Common Vulnerabilities and Exposures identifier (e.g., CVE-2004-2687). You can use this to search for exploit code and detailed information online.
     - CVSS Score: This is the severity score (e.g., 9.8 (Critical)). This tells you how dangerous the vulnerability is.
     - Location: The specific IP address and port where the vulnerability was found (e.g., 192.168.56.102/tcp/3632).
     - Solution/Summary: GVM provides a description of the vulnerability and often gives recommendations on how to fix it.
4. Export the Results: You can export the entire report for documentation.
   - From the Tasks screen, select your completed scan.
   - Click the Download icon.
   - Choose a format like PDF or XML. The XML format is excellent for importing into other tools like Dradis CE for report generation.



Kali 2.0 ... TCP

Target for immediate scan of 2025-12-18 11:16:40 ... TCP

Target for immediate scan of 2025-12-18 11:19:27 ... TCP

Target for immediate scan of 2025-12-18 11:30:13 ... TCP

Target for immediate scan of 2025-12-18 11:30:22 ... TCP

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

New Target

Name: Metasploitable 3

Comment:

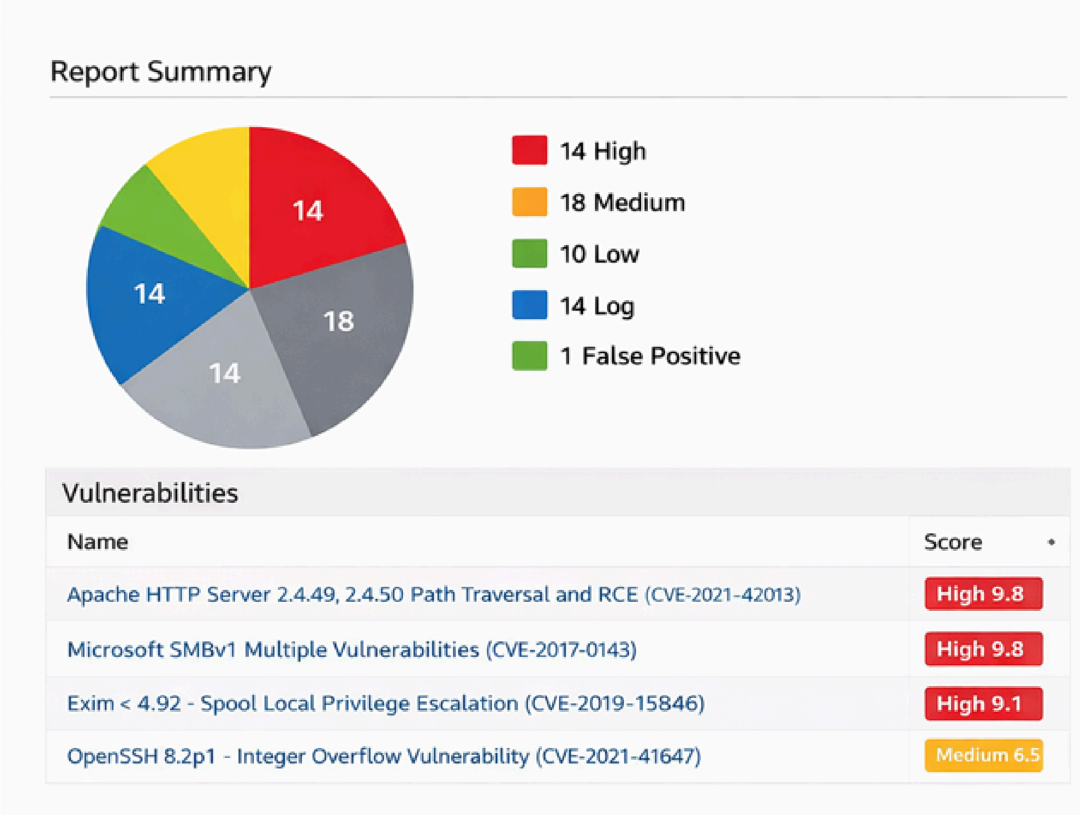
Hosts
● Manual 192.168.132.133
○ From file

Exclude Hosts
● Manual
○ From file

Allow simultaneous scanning via multiple IPs
● Yes ○ No

Port List: All IANA assigned TCP

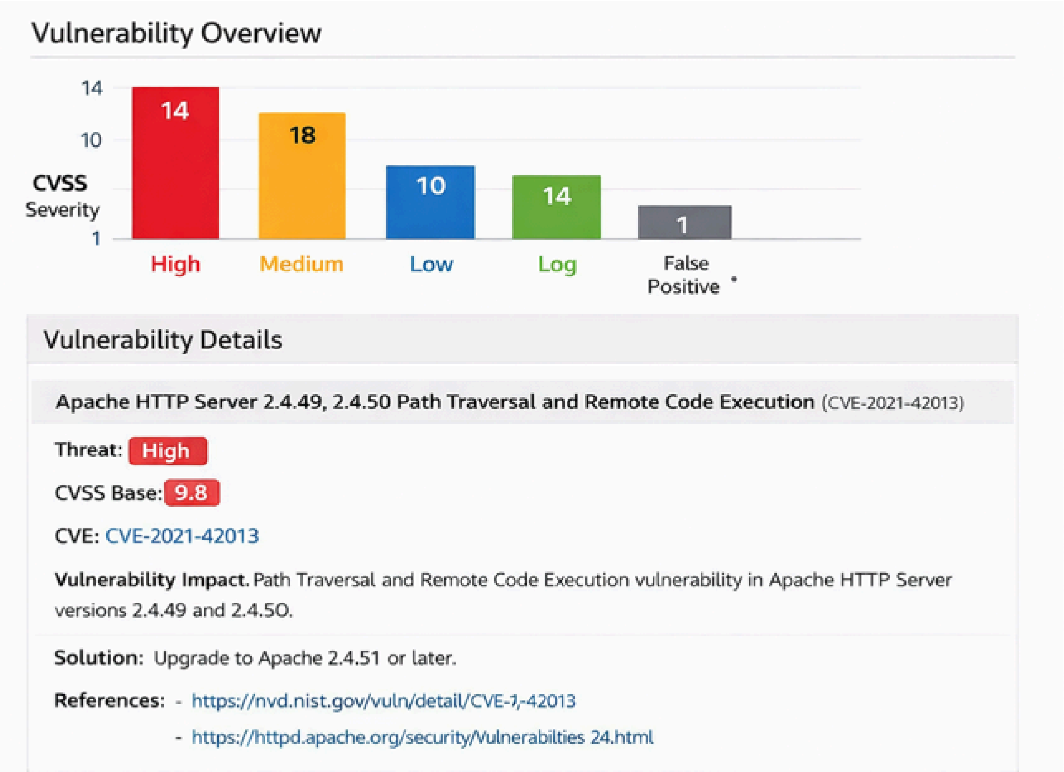
Alive Test
● Use Scan Config Default ○ Consider Hosts as Alive ○ Custom



Report Summary

Vulnerabilities

Name	Score
Apache HTTP Server 2.4.49, 2.4.50 Path Traversal and RCE (CVE-2021-42013)	High 9.8
Microsoft SMBv1 Multiple Vulnerabilities (CVE-2017-0143)	High 9.8
Exim < 4.92 - Spool Local Privilege Escalation (CVE-2019-15846)	High 9.1
OpenSSH 8.2p1 - Integer Overflow Vulnerability (CVE-2021-41647)	Medium 6.5



Vulnerability Overview

Vulnerability Details

Apache HTTP Server 2.4.49, 2.4.50 Path Traversal and Remote Code Execution (CVE-2021-42013)

Threat: High

CVSS Base: 9.8

CVE: CVE-2021-42013

Vulnerability Impact. Path Traversal and Remote Code Execution vulnerability in Apache HTTP Server versions 2.4.49 and 2.4.5O.

Solution: Upgrade to Apache 2.4.51 or later.

References: - https://nvd.nist.gov/vuln/detail/CVE-1,-42013
- https://httpd.apache.org/security/Vulnerabilties 24.html

Using Nikto for Web Scanning

While OpenVAS is great for network-wide scanning, Nikto is a specialized web server scanner. It's fast and focuses on common web server misconfigurations and dangerous files.

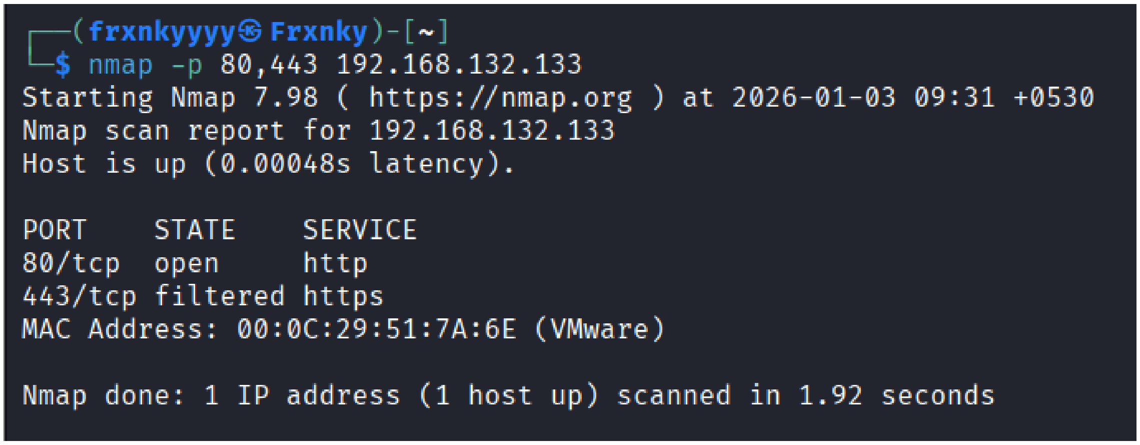
1. Open a Terminal in Kali.
2. Run the Scan: You need to point Nikto at the Metasploitable web server, which is running on port 80.
3. bash
4. nikto -h http://[IP_Address_of_Metasploitable]

- e.g., nikto -h http://192.168.132.133 (In my case)

5. Analyze the Output: Nikto will print its findings directly to the terminal. We will see items like:
    - "OSVDB-[ID]: /admin/: Directory indexing found"
    - "OSVDB-[ID]: /config/: Configuration file may contain sensitive information"
    - "OSVDB-[ID]: Apache/2.2.8 appears to be outdated"

These findings from OpenVAS and Nikto now give us a clear target list. We can take a high-severity finding, like the "DistCC Daemon Command Execution" (CVE-2004-2687), and use it as your next objective: find and run the Metasploit exploit for it against your Metasploitable VM.

- Check for the open port 80

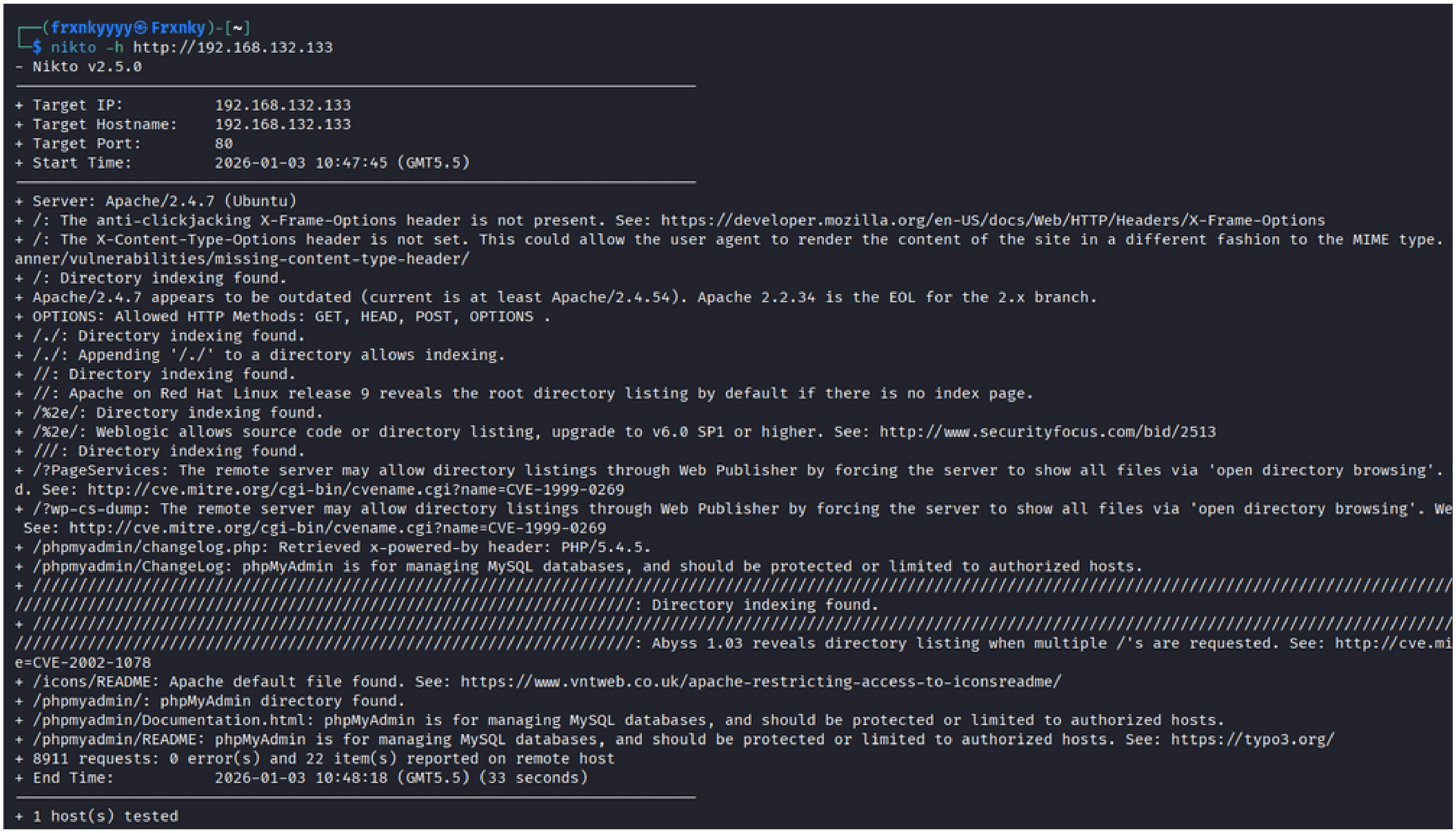


```
┌──(frxnkyyyy㉿Frxnky)-[~]
└─$ nmap -p 80,443 192.168.132.133
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-03 09:31 +0530
Nmap scan report for 192.168.132.133
Host is up (0.00048s latency).

PORT    STATE    SERVICE
80/tcp  open     http
443/tcp filtered https
MAC Address: 00:0C:29:51:7A:6E (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.92 seconds
```

- Results



```
┌──(frxnkyyyy㉿Frxnky)-[~]
└─$ nikto -h http://192.168.132.133
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.132.133
+ Target Hostname:    192.168.132.133
+ Target Port:        80
+ Start Time:         2026-01-03 10:47:45 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
anner/vulnerabilities/missing-content-type-header/
+ /: Directory indexing found.
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /./: Directory indexing found.
+ /./: Appending '/./' to a directory allows indexing.
+ //: Directory indexing found.
+ //: Apache on Red Hat Linux release 9 reveals the root directory listing by default if there is no index page.
+ /%2e/: Directory indexing found.
+ /%2e/: Weblogic allows source code or directory listing, upgrade to v6.0 SP1 or higher. See: http://www.securityfocus.com/bid/2513
+ ///: Directory indexing found.
+ /?PageServices: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'.
d. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /?wp-cs-dump: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. We
 See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /phpmyadmin/changelog.php: Retrieved x-powered-by header: PHP/5.4.5.
+ /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////: Directory indexing found.
+ ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////: Abyss 1.03 reveals directory listing when multiple /'s are requested. See: http://cve.mi
e=CVE-2002-1078
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpmyadmin/: phpMyAdmin directory found.
+ /phpmyadmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpmyadmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
+ 8911 requests: 0 error(s) and 22 item(s) reported on remote host
+ End Time:           2026-01-03 10:48:18 (GMT5.5) (33 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

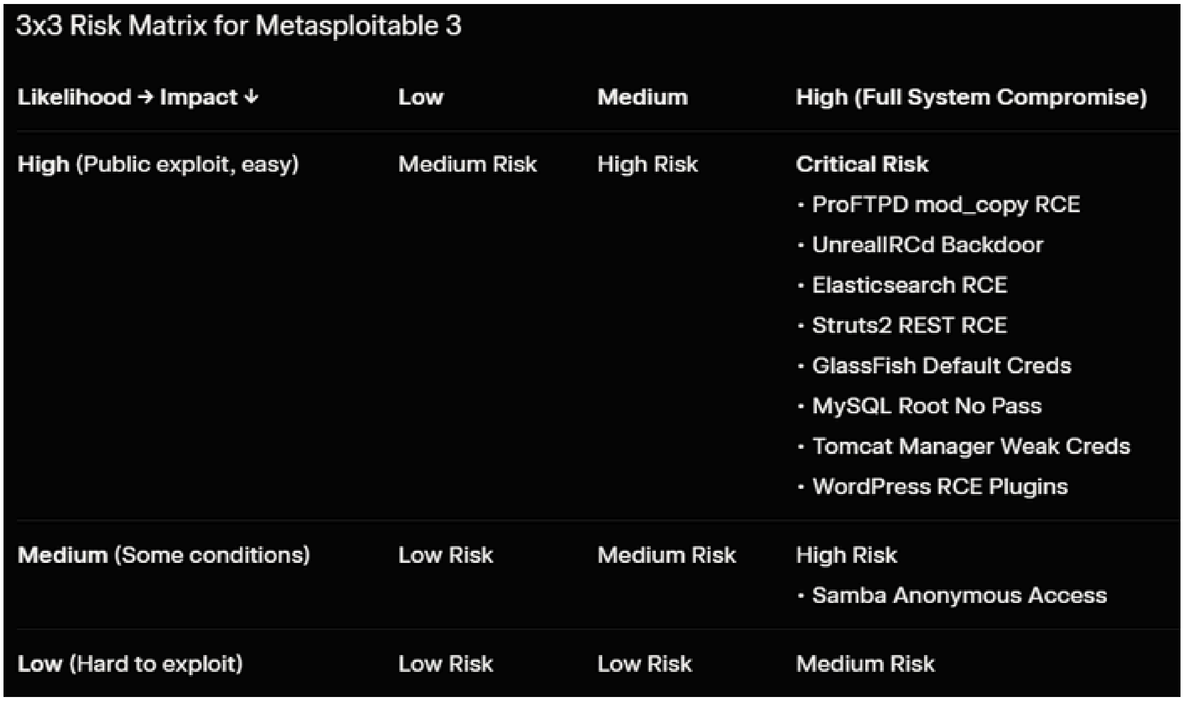
Practical Assessment: Documenting Findings for Metasploitable 3
Objective: Practice professional documentation by recording real findings from your Metasploitable 3 lab, using spreadsheets for vulnerability tracking and embedding screenshots as evidence.

Link for the Google spreadsheet
https://docs.google.com/spreadsheets/d/12sHtspAyCwK-AgS_4tBMbDCSNHVctRthJGIBKnq9f2Q/edit?usp=sharing

Practice Risk Assessment

Prioritize Risks Using a 3x3 Risk Matrix
We use a 3x3 Risk Matrix combining Likelihood and Impact.

- Likelihood (of exploitation in this lab environment):
  - High: Public exploit available, no authentication required, reliable.
  - Medium: Requires some interaction or low-privilege access.
  - Low: Complex, unreliable, or requires local access.
- Impact (if exploited):
  - High: Full system compromise (root/remote code execution).
  - Medium: Data disclosure or partial control.
  - Low: Limited or informational only.



3x3 Risk Matrix for Metasploitable 3

Likelihood → Impact ↓	Low	Medium	High (Full System Compromise)
High (Public exploit, easy)	Medium Risk	High Risk	Critical Risk • ProFTPD mod_copy RCE • UnrealIRCd Backdoor • Elasticsearch RCE • Struts2 REST RCE • GlassFish Default Creds • MySQL Root No Pass • Tomcat Manager Weak Creds • WordPress RCE Plugins
Medium (Some conditions)	Low Risk	Medium Risk	High Risk • Samba Anonymous Access
Low (Hard to exploit)	Low Risk	Low Risk	Medium Risk

Summary of Prioritization:

- Critical Risk (Immediate Action): 8 vulnerabilities — all allow remote unauthenticated code execution or full admin access.
- High Risk: 1–2 (e.g., Samba information disclosure leading to further attacks).
- Medium/Low Risk: None in core findings (this VM is designed to be highly exploitable).