

Final VAPT Report Template



PREPARED BY: Anto Sebastine Jebin

Submitted To: Tech Next LLC's

Submission Date: 09/09/2025

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
MITRE ATT&CK	4
HIGH LEVEL ASSESSMENT OVERVIEW	
SCOPE	5
• Project Scope	4
• Part 1: Zero Bank's Network	4
• Part 2: VSFTP Very Secure FTP Daaemon	4
• Part 3: DVWA and Mutillidae	
• Part 4: Active Directory	Error! Bookmark not defined.
NETWORK INFORMATION	
TESTING METHODOLOGY	9
CLASSIFICATION DEFINITIONS	11
• Risk Classifications	7
• Exploitation Likelihood Classifications	7
• Business Impact Classifications	8
• Remediation Difficulty Classifications	8
ASSESSMENT FINDINGS	
• TechNext_Internal_Network_Scan_Sep2025 -Nessus	
VULNERABILITY FINDING	
SUGGESTED REMEDIATION	
GENERAL RECOMMENDATIONS	
APPENDIX A - TOOLS USED	12
APPENDIX B - ENGAGEMENT INFORMATION	13
• Client Information	13
• Version Information	13
• Contact Information	13

EXECUTIVE SUMMARY

TechNest LLC's performed a security assessment of the internal corporate network of Zero Bank's on 09/09/2025. IT Team's penetration test simulated an attack from an external threat actor attempting to gain access to systems within the Zero Bank's corporate network. The purpose of this assessment was to discover and identify vulnerabilities in Zero Bank's infrastructure and suggest methods to remediate the vulnerabilities. IT Team identified a total of 20 vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

CRITICAL	HIGH	MEDIUM	LOW
2	6	8	4

The highest severity vulnerabilities give potential attackers the opportunity to intercept and poison NetBIOS and LLMNR name resolution requests, capture NTLMv2 authentication hashes, and perform SMB relay attacks to compromise Active Directory service accounts. By exploiting these weaknesses, adversaries could escalate privileges to domain-admin level, gain unrestricted access to sensitive file shares, and tamper with authentication mechanisms. In addition, flaws such as remote code execution, the vsftpd backdoor, and SQL injection expose critical systems and databases to unauthorized command execution, backdoor access, and direct manipulation of customer and financial data. Through these attack paths, an attacker could move laterally across the network, harvest further credentials, implant persistence within key systems, and exfiltrate or alter business-critical information—placing the confidentiality, integrity, and availability of the organization's environment at severe risk. In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope. Any changes made to the environment during the period of testing may affect the results of the assessment.



HIGH LEVEL ASSESSMENT OVERVIEW

Observed Security Strengths

TechNest LLC's security identified the following strengths in Zero Bank's network which greatly increases the security of the network. Zero Bank should continue to monitor these controls to ensure they remain effective.

Network and Service Controls:

The network has disabled unnecessary and high-risk services, reducing the attack surface and minimizing exposure to potential threats.

Endpoint Protection:

All endpoints have Windows Defender properly configured, providing real-time protection against malware and other malicious activity.

Centralized Monitoring (SIEM):

A SIEM solution (Wazuh) is deployed to collect logs, correlate events, and detect suspicious activity proactively.

Access Control:

Strong user authentication and access policies are enforced, reducing the risk of unauthorized access to critical systems.

Patch Management:

Systems and servers are regularly updated and patched to mitigate exposure to known vulnerabilities.

Network Segmentation:

Critical servers and sensitive applications are logically segregated from general user networks, limiting lateral movement in case of compromise.



Areas for Improvement

TechNest security Team recommends Secure Solutions 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 Secure Solutions's information systems and/or reduce the impact of a successful attack.

Short Term Recommendations

TechNest's security team recommends Secure Solutions take the following actions as soon as possible to minimize business risk.

Backdoor Mitigation:

Remove or secure the backdoored vsftpd service to eliminate unauthorized access points.

Web Application Security:

Apply input validation and fix SQL injection vulnerabilities in web applications to protect sensitive customer and financial data.

Correct Cross-Site Scripting (XSS) vulnerabilities in web applications to prevent session hijacking and unauthorized data access.

Network Security:

Implement mitigations for LLMNR/NBT-NS poisoning and SMB relay attacks, such as disabling unnecessary protocols and enforcing strong authentication.

Access and Credential Management:

Enforce strict account policies and credential protection to prevent compromise of privileged accounts.

Long Term Recommendations

TechNest's security team recommends the following actions be taken over the next 6-12 months to fix hard-to-remediate issues that do not pose an urgent risk to the business.



Monitoring and Detection:

- Strengthen network monitoring and alerting to detect suspicious authentication attempts or unusual traffic patterns.
- Conduct regular penetration testing and vulnerability assessments to proactively identify emerging threats.
- Implement advanced log correlation and anomaly detection within the SIEM to identify lateral movement and suspicious privilege escalation.

Access and Privilege Management:

- Implement comprehensive privilege management and enforce least-privilege policies across all users and service accounts.
- Review and enhance network segmentation and access controls to isolate critical systems and sensitive applications.
- Introduce multi-factor authentication (MFA) for all high-privilege accounts and remote access points.

Web Application and Database Security:

- Conduct code reviews and security testing for all web applications to prevent SQL injection, XSS, and other injection-based attacks.
- Implement a secure development lifecycle (SDLC) and regular vulnerability scanning for all in-house applications.
- Deploy web application firewalls (WAF) to monitor and block malicious HTTP/HTTPS requests targeting web applications.

Endpoint and Infrastructure Hardening:

- Regularly review system configurations and disable unnecessary services to reduce attack surfaces.
- Ensure consistent patching of servers, endpoints, and network devices for all known vulnerabilities.
- Enforce configuration baselines for endpoints and servers using automated compliance tools.

SCOPE

Project Scope

All testing was performed within the boundaries defined in the Request for Proposal (RFP) and official communications. The in-scope items were:

- Zero Bank's internal network and Active Directory
- Windows client systems
- Application server (FTP and web services: VSFTP, DVWA)
- Mutillidae web application
- Mutillidae database system

Network Information

Network	Machine	Purpose
192.168.137.10	Kali Linux	Attacker Machine
192.168.137.67	Windows 10	Windows Machine
192.168.137.58	Windows Server	Active Directory
192.168.137.20	Metasploitable	Web Application
192.168.137.0/24	Network Address	Defines the entire network/s

TESTING METHODOLOGY

TechNest's testing methodology was divided into five phases: Scope Definition, Reconnaissance, Vulnerability Analysis, Exploitation, and Reporting with Remediation. During reconnaissance, information about Zero Bank's network systems was collected. TechNest's security team performed port scanning, service enumeration, and OS fingerprinting to refine target information and assess potential attack surfaces. Next, a targeted assessment was conducted, where the team simulated an attacker attempting to exploit vulnerabilities within the environment. Evidence of each vulnerability was gathered during this phase, while ensuring that testing was carried out in a controlled manner without disrupting normal business operations.

Testing Methodology

1. Planning

- Defined the scope to include Zero Bank's internal network, Active Directory, Windows client systems, FTP services, web applications (DVWA, Mutillidae), and associated databases.
- Selected a hybrid testing approach using both automated tools (e.g., Nmap, vulnerability scanners) and manual exploitation techniques.
- Established testing rules of engagement to avoid disrupting business operations.

2. Target Acquisition


- Conducted network scanning to identify live hosts and open ports.
- Fingerprinted operating systems (Windows and Linux) and enumerated services such as SMB, FTP, and HTTP.
- Mapped potential attack surfaces, including SAM/SMB services, vsftpd service, and multiple web applications.

3. Pre-Exploitation

- Matched identified services to known vulnerabilities: vsftpd backdoor, SMB relay, SQL injection, XSS, and directory traversal.
- Validated opportunities for credential compromise through dictionary attacks and NTLMv2 hash capture.
- Analyzed AD and SAM configurations for privilege escalation risks.

4. Target Engagement

- Exploited the vsftpd backdoor to gain shell-level access.
- Carried out SQL injection on the Mutillidae database to extract sensitive data.

- 
- Executed LLMNR/NBT-NS poisoning with Responder to capture NTLMv2 hashes and simulate SMB relay attacks.
 - Performed dictionary-based brute force against Windows accounts.
 - Tested web applications for XSS and directory traversal to demonstrate unauthorized data access.

5. Post-Exploitation

- Demonstrated privilege escalation on compromised systems using weak configurations and credential reuse.
- Simulated lateral movement within the network via compromised AD accounts.
- Evaluated risks of persistence through malware-based session hijacking.
- Highlighted potential for sensitive file share access, customer/financial data exfiltration, and AD object manipulation.

6. Documentation

- Captured evidence of exploitation (screenshots, logs, hashes, extracted data).
- Assigned severity levels (Critical, High, Medium, Low) mapped to CVSS scores.
- Provided actionable remediation recommendations (disable LLMNR, patch SQL injection, secure FTP, enforce MFA, strengthen AD policies).

The following image is a graphical representation of this methodology.



CLASSIFICATION DEFINITIONS

Risk Classifications

Level	Score	Description
Critical	10	The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed.
High	7-9	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.
Medium	4-6	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.

Low	1-3	The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.
Informational	0	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
Likely	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.
Possible	Exploitation methods are well-known, may be performed using public tools, but require configuration. Understanding of the underlying system is required for successful exploitation.
Unlikely	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
Major	Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage.
Moderate	Successful exploitation may cause significant disruptions to non-critical business functions.
Minor	Successful exploitation may affect few users, without causing much disruption to routine business functions.

Remediation Difficulty Classifications

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

ASSESSMENT FINDINGS

Number	Finding	Risk Score	Risk
1	Exploitable MS17-010 (Eternal Blue) Vulnerability In Windows	10	High
2	VSFTPD Misconfiguration	9	High
3	Weak Password Policy in Active Directory	9	High
4	No WAF (Web Application Firewall)	9	High
5	Sql Vulnerability in Input and url field	7	High
6	Unpatched Software in Mutillidae	7	High
7	Outdated Softwares	7	High
8	Outdated Protocols / network services	7	High
9	Cross-Site Request Forgery (CSRF) in DVWA	7	Medium
10	File Upload Vulnerability	6	Medium

11	Insecure File Sharing on High-Value System	6	Medium
12	Limited Logging on Sensitive Systems	5	Medium
13	Misconfigured Active Directory Permissions	5	Medium
14	Multiple service Vulnerabilities	5	Medium
15	Cross-Site Scripting (XSS) Vulnerability in Mutillidae	4	Medium
16	Lack of Multi-Factor Authentication for Admin Accounts	4	Medium
17	Use of Default Credentials	3	Low
18	Information Disclosure via Error Messages	2	Low
19	Unused Open Ports on Critical Systems ¹	1	Low
20	Lack of Security Awareness Training	1	Low

1 - Example Vulnerability Finding

MS17-010 (Eternal Blue) Vulnerability

HIGH RISK (10/10)	
Exploitation Likelihood	Possible
Business Impact	Severe
Remediation Difficulty	Easy

Security Implications

This vulnerability allows remote code execution on the affected host through the SMBv1 service, enabling attackers to gain full control without authentication. This finding is very important because it can be weaponized (as seen in WannaCry and NotPetya) to spread rapidly across networks, potentially crippling operations and destroying the entire business if left unchecked.

Analysis

The target system was found vulnerable to **MS17-010**, a critical Windows SMBv1 vulnerability (CVE-2017-0144) exploited by the **EternalBlue exploit**. This flaw exists because SMBv1 improperly handles specially crafted packets, allowing an unauthenticated remote attacker to execute arbitrary code on the system. Exploitation provides **full system compromise**, often leading to malware deployment, data theft, or ransomware outbreaks.

EVIDENCE :

```
└─$ nmap -p 445 --script smb-vuln 192.168.137.67
Starting Nmap 7.93 ( https://nmap.org ) at 2025-09-10 11:09 EDT
Nmap scan report for 192.168.137.67
Host is up (0.0020s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms17-010:
|  VULNERABLE:
|    Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|    State: VULNERABLE
|    IDs: CVE:CVE-2017-0143
|    Risk factor: HIGH
|    A critical remote code execution vulnerability exists in Microsoft SMBv1
|    servers (ms17-010).
|
|    Disclosure date: 2017-03-14
|    References:
|    https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|    https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED

Nmap done: 1 IP address (1 host up) scanned in 19.07 seconds
```

Figure 2.3.1: Eternal Blue Vulnerability

2 -Vulnerability Finding Insecure Configuration in VSFTPD

HIGH RISK (8/10)	
Exploitation Likelihood	Possible
Business Impact	Severe
Remediation Difficulty	Easy

Security Implications

This vulnerability allows attackers to exploit a backdoor in the VSFTPD 2.3.4 service, enabling them to gain an unauthorized shell on the affected host. The issue arises from a maliciously altered version of the software distributed in 2011, where sending a specially crafted username triggers the backdoor. This finding is very important because successful exploitation grants attackers remote shell access, which can lead to complete system compromise, unauthorized data access, and the potential use of the compromised host as a pivot point to attack other systems.

Analysis

The target system was found running **VSFTPD version 2.3.4**, which is known to be backdoored. When an attacker connects to the FTP service and supplies a username ending with the string `;`, the server opens a shell on TCP port 6200. This insecure configuration permits unauthenticated remote access and control of the system. Exploitation can result in the attacker bypassing authentication, gaining remote command execution, stealing sensitive data, and installing malicious software. Due to the ease of exploitation and high impact, this vulnerability poses a severe risk to the affected system and the overall network security posture.

Evidence:

```
(stack@kali)-[~]
$ nmap -sV -A 192.168.137.20
Starting Nmap 7.93 ( https://nmap.org ) at 2025-09-10 11:22 EDT
Nmap scan report for 192.168.137.20
Host is up (0.025s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)|
| ftp-syst:
|   STAT:
|   FTP server status:
|   | Connected to 192.168.137.10
|   | Logged in as ftp
|   | TYPE: ASCII
|   | No session bandwidth limit
|   | Session timeout in seconds is 300
|   | Control connection is plain text
|   | Data connections will be plain text
|   | vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
|   2048 5656240f211ddea72bae61b1243de8f3 (RSA)
23/tcp    open  telnet       Linux telnetd
```

Figure 2.3.2: VSFTPD V 2.3.4

```

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshcd
513/tcp   open  login?       Netkit rshd
514/tcp   open  shell        GNU Classpath grmiregistry
1099/tcp  open  java-rmi     Bash shell (**BACKDOOR**; root shell)
1524/tcp  open  bindshell    2-4 (RPC #100003)
2049/tcp  open  nfs         MySQL 5.0.51a-3ubuntu5
3306/tcp  open  mysql        PostgreSQL DB 8.3.0 - 8.3.7
5432/tcp  open  postgresql   VNC (protocol 3.3)
5900/tcp  open  vnc          (access denied)
6000/tcp  open  X11          UnrealIRCd
6667/tcp  open  irc          Apache Jserv (Protocol v1.3)
8009/tcp  open  ajp13        Apache Tomcat/Coyote JSP engine 1.1
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, metasploitable-web, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/
:linux:linux_kernel

```

```
msf6 > search vsftpd 2.3.4
```

```
Matching Modules
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	No	VSFTPD v2.3.4 Backdoor or Command Execution

Interact with a module by name or index. For example `info 0`, `use 0` or `use exploit/unix/ftp/vsftpd_234_backdoor`

```
msf6 >
```

```
stack@kali: ~ x
```

```
[*] 192.168.137.20:21 - USER: 331 Please specify the password.
```

```
[*] Exploit completed, but no session was created.
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
```

```
[*] 192.168.137.20:21 - The port used by the backdoor bind listener is already open
```

```
[+] 192.168.137.20:21 - UID: uid=0(root) gid=0(root)
```

```
[*] Found shell.
```

```
[*] Command shell session 1 opened (192.168.137.10:35525 → 192.168.137.20:6200) at 2025-09-10 17:19 -0400
```

```
whoami
```

```
root
```

```
ls
```

```
bin
```

```
boot
```

```
cdrom
```

```
dev
```

```
etc
```

FILE TRANS

3 – Sql Vulnerability In Metasploitable

HIGH RISK (8/10)	
Exploitation Likelihood	Possible
Business Impact	Severe
Remediation Difficulty	Easy

Security Implications

This vulnerability allows attackers to inject malicious SQL statements into the application's database queries, potentially bypassing authentication, retrieving sensitive information, or modifying/deleting data. This finding is very important because it can expose customer records, credentials, and financial data, ultimately compromising the integrity and confidentiality of the entire database. If left unchecked, it could destroy trust and business continuity.

Analysis

The target web application was found vulnerable to **SQL Injection** due to improper validation of user-supplied input in database queries. By manipulating input fields such as login forms, search boxes, or URL parameters, an attacker can alter the SQL query logic. This flaw allows unauthorized database access, potentially exposing sensitive data such as usernames, passwords, credit card information, and business records. In advanced attacks, SQL Injection can be used to escalate privileges, drop database tables, or gain remote code execution on the underlying server.

Evidence :

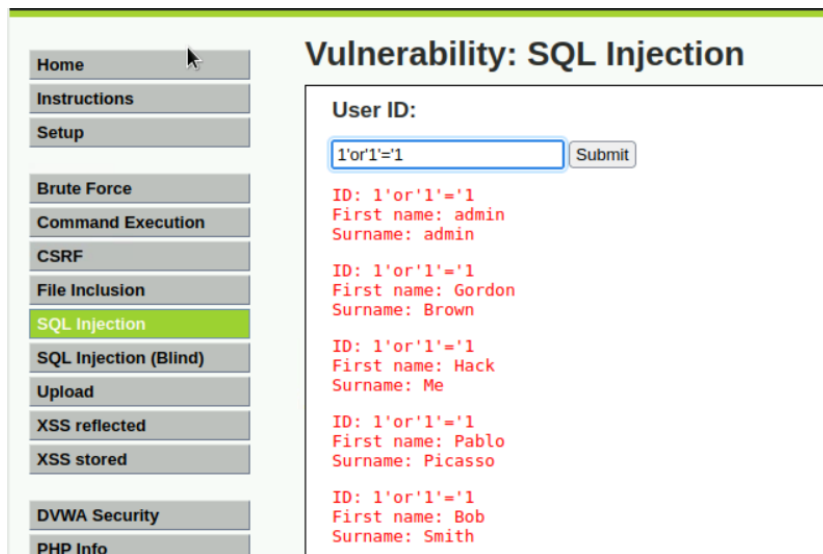


Figure 2.3.3: Sql Injection

```
[11:50:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL > 4.1
[11:50:43] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[11:50:43] [INFO] fetched data logged to text files under '/home/stack/.local/share/sqlmap/output/192.168.137.20'
[11:50:43] [WARNING] your sqlmap version is outdated

[*] ending @ 11:50:43 /2025-09-06/
```

3 – XSS Vulnerability

HIGH RISK (8/10)	
Exploitation Likelihood	Possible
Business Impact	Severe
Remediation Difficulty	Easy

Security Implications

This vulnerability allows attackers to inject malicious scripts into web pages that are viewed by other users. When exploited, attackers can steal cookies, hijack sessions, redirect users to malicious sites, or perform actions on behalf of the victim. This finding is very important because it directly affects end users and can lead to large-scale account compromise, data theft, and loss of customer trust.

Analysis

The target application was found vulnerable to **Cross-Site Scripting (XSS)** because it does not properly sanitize user-supplied input before reflecting it in the web page output. Attackers can exploit this by injecting JavaScript payloads into input fields (e.g., search forms, comments, or URL parameters). Once executed in a victim's browser, the malicious script can steal session tokens, perform phishing attacks, or manipulate web content. In persistent XSS cases, the malicious payload remains stored in the database and executes every time a user visits the affected page, multiplying the impact.

EVIDENCE :

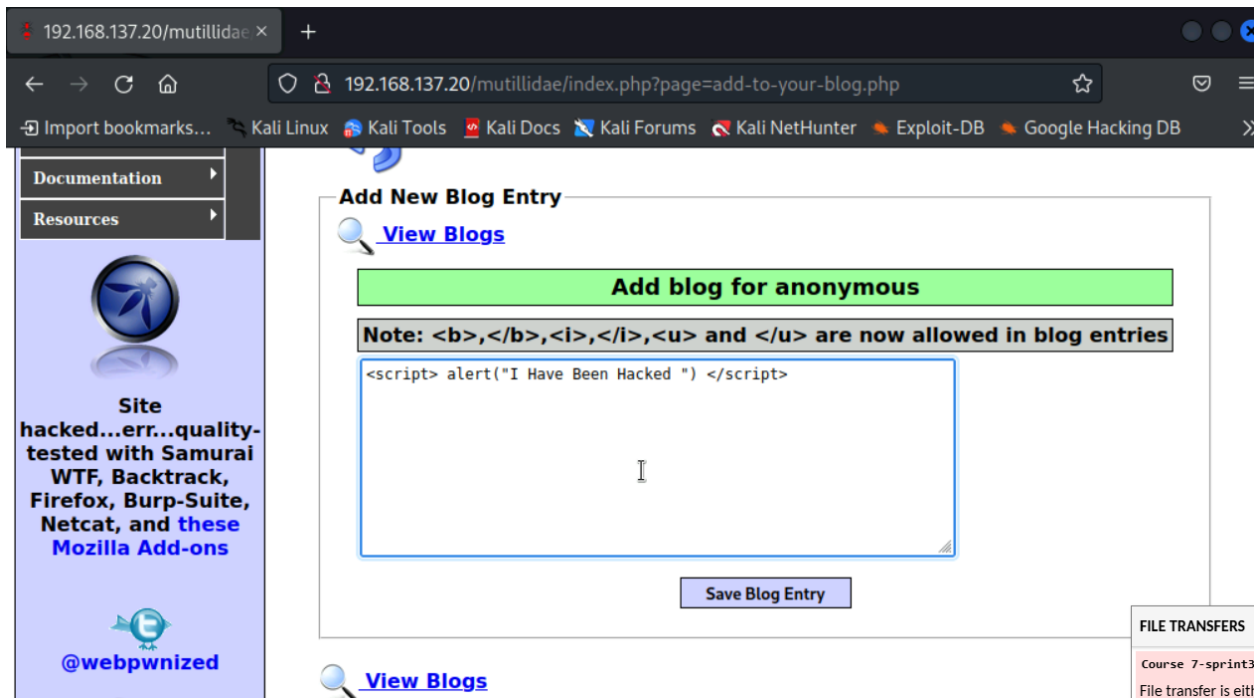
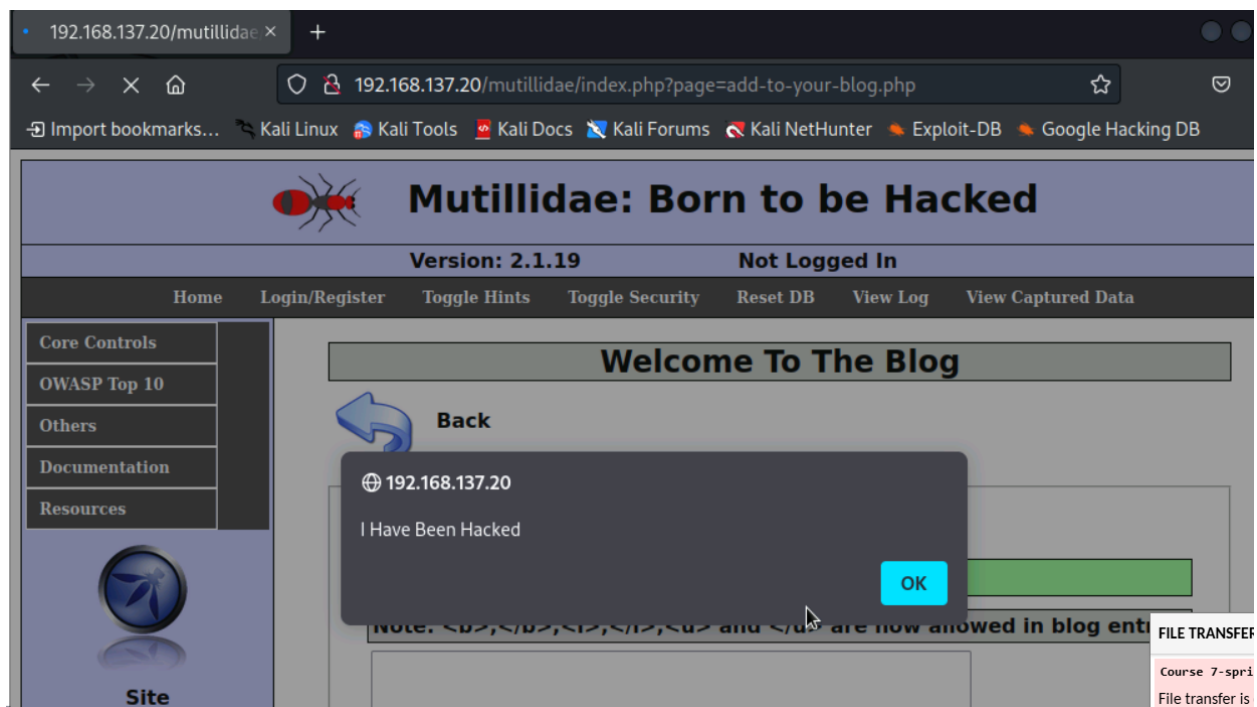


Figure 2.3.4: XSS Injection



SUGGESTED REMEDIATION

Recommendations

To strengthen the overall security posture of the environment, the following actions are recommended for the identified vulnerabilities.

Part 1: Zero Bank Network

Patch Management

- Regularly update all operating systems and applications, focusing on Windows SMB services to mitigate vulnerabilities like MS17-010.

Access Control

- Implement strict role-based access so only authorized administrators can access critical systems. Regular users should have minimum permissions necessary for their roles.

Network Segmentation

- Segment the network to restrict communication between departments, reducing the risk of lateral movement by attackers.

Incident Response Plan

- Develop, test, and update an incident response plan to quickly detect, contain, and recover from security incidents.



Part 2: Metasploitable (VSFTP and DVWA)

Configuration Hardening

- Disable anonymous FTP access on VSFTP and enforce proper account-based restrictions.

Web Application Security

- Deploy security headers such as Content Security Policy and X-Content-Type-Options to reduce exposure to web attacks.

Input Validation

- Enforce strict input validation and sanitization to prevent SQL injection and other injection attacks.

Part 3: Mutillidae

Regular Security Testing

- Conduct regular security assessments including penetration testing, vulnerability scanning, and code reviews to detect and remediate issues promptly.

Sensitive Data Protection

- Encrypt sensitive data both in transit and at rest to prevent unauthorized access.

User Authentication and Session Management

- Implement strong password policies and multi-factor authentication for all user accounts.

Part 4: TechNest LLC's Network

Monitoring and Logging

- Enhance monitoring and logging practices to detect suspicious activities and support incident response.

Password Management

- Enforce complex password requirements and encourage the use of password managers to secure credentials.

Training and Awareness

- Conduct ongoing security awareness training to help employees recognize phishing attempts and social engineering tactics.

APPENDIX A - TOOLS USED

TOOL	DESCRIPTION
BurpSuite Community Edition	Test web applications for security weaknesses.
Metasploit	Framework for finding and exploiting system vulnerabilities.
Nmap	Scan networks to discover devices and open ports.
Nessus	Identify vulnerabilities and misconfigurations in systems.
SQLMap	Detect and exploit SQL injection vulnerabilities.
Wireshark	Capture and analyze network traffic in real-time.
Aircrack-ng	Assess Wi-Fi network security and test password strength.
Wazuh	Monitor security and compliance across systems.
Hydra	Perform brute-force attacks to test password strength.
John the Ripper	Test and crack passwords to assess strength.
PostgreSQL Client Tools	Manage and interact with PostgreSQL databases.

Table A.1: Tools used during assessment

APPENDIX B - ENGAGEMENT INFORMATION

Client Information

Client	Secure Solutions
Primary Contact	Chief Information Officer
Approvers	The following people are authorized to change the scope of engagement and modify the terms of the engagement <ul style="list-style-type: none">• Chief Information OFFICER• Chief Information Security Officer

Version Information

Version	Date	Description
1.0	09/09/2025	Initial report to client

Contact Information

Name	Anto Jebin
Address	1001 Fake Street, Gotham, NY 11201
Phone	555-185-1782
Email	antojebin@gmail.com