**Network Vulnerability Assessment Report**

**Final Project - DEPI**

**Prepared for:**

DEPI (Digital Egypt Penetration Initiative) - Round 1 Final Project Assessment

**Prepared by:**

[Wael Ashraf Azouz Ali Eldabah ]

**Group Code:**

CAI1\_ISS5\_S3e

**Team Members:**

Wael Ashraf, Mahmoud Ehab, Zeyad Assem, Haidy Karam, Nada Mohamed

**Date:**

October 21, 2024

**Version:**

1.0

**Disclaimer**

This penetration testing report provides a point-in-time assessment of the security of 8 VulnHub machines. The findings and recommendations are based on information gathered during the assessment and do not account for any changes made to the systems afterward. Due to time constraints, not all security controls could be evaluated. The assessment prioritized the most vulnerable areas that an attacker could exploit. It is recommended to conduct periodic penetration tests to ensure ongoing security effectiveness.

**Assessment Overview**

This penetration testing assessment aimed to identify and document vulnerabilities within 8 VulnHub machines. Comprehensive testing was conducted to uncover potential security risks, including network misconfigurations, software flaws, and insecure access controls. The analysis provides detailed explanations of how these vulnerabilities can be exploited and offers actionable mitigation strategies to strengthen system security.

The assessment followed a structured methodology, including pre-engagement planning, reconnaissance, vulnerability scanning, exploitation, post-exploitation evaluation, and detailed reporting. This ensured a thorough evaluation of the target systems' security posture.

**Key Deliverables**

* A detailed vulnerability report outlining identified vulnerabilities, exploitation methods, and potential impact.
* Actionable mitigation strategies to address each discovered vulnerability and enhance system security.
* A comprehensive analysis of potential attack vectors and their implications.

**Severity Ratings**

| Severity | CVSS v3 Score Range | Definition |
| --- | --- | --- |
| Critical | 9.0 - 10.0 | Exploitation is straightforward and usually results in system-level compromise. Immediate patching is advised. |
| High | 7.0 - 8.9 | Exploitation is more difficult but could cause elevated privileges and potentially loss of data or downtime. Patching should be prioritized. |
| Moderate | 4.0 - 6.9 | Vulnerabilities exist but are not easily exploitable or require extra steps. |
| Low | 0.1 - 3.9 | Vulnerabilities are non-exploitable but could reduce the organization's attack surface. |
| Informational | N/A | No vulnerability exists. Provides additional information or observations. |

**CVSS (Common Vulnerability Scoring System) v3**

CVSS v3 is a standardized framework for assessing the severity of software vulnerabilities. It provides a numerical score that helps organizations prioritize security risks. For detailed information about CVSS v3, refer to the [CVSS v3 Calculator](https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator).

**Scope**

The following VulnHub machines were included in the scope of this assessment:

* Kioptrix Level 1 (192.168.187.133)
* Kioptrix Level 2 (192.168.50.132)
* Kioptrix Level 4 (192.168.100.84)
* Lampiao (192.168.100.70)
* Metasploitable 1 (192.168.8.134)
* Metasploitable 2 (192.168.50.133)
* SkyTower 1 (10.0.2.9)
* Stapler 1 (10.0.2.10)

**Scope Exclusions**

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks were excluded from the scope of this assessment.

**Executive Summary**

This penetration testing assessment aimed to identify and evaluate security vulnerabilities within 8 VulnHub machines. Through comprehensive testing, we uncovered potential risks, including network misconfigurations, software flaws, and insecure access controls. Our analysis provides detailed explanations of how these vulnerabilities could be exploited and offers actionable mitigation strategies to enhance system security. The following sections provide more detail regarding the findings.

**Methodology**

The assessment followed a structured penetration testing approach that involved both automated scanning and manual exploitation. The process included the following steps:  
- \*\*Information Gathering:\*\* Scanning for open ports and services, gathering metadata.  
- \*\*Vulnerability Scanning:\*\* Automated tools (e.g., Nmap, Burp Suite) were used to detect vulnerabilities.  
- \*\*Exploitation:\*\* Attempts were made to exploit weaknesses using Metasploit and other frameworks.  
- \*\*Post-Exploitation:\*\* Evaluating the impact of successful exploitation, privilege escalation, and data access.  
- \*\*Remediation and Reporting:\*\* Providing actionable recommendations to mitigate risks.

**Tools Used**

Several tools were utilized to conduct the penetration test and gather data, including:  
- \*\*Metasploit Framework:\*\* Exploitation and post-exploitation.  
- \*\*Nmap:\*\* Network scanning and enumeration.  
- \*\*Burp Suite:\*\* Web application vulnerability scanning.  
- \*\*WPScan:\*\* WordPress security scanner.  
- \*\*Hydra:\*\* Brute-force attack testing.  
- \*\*Dirbuster:\*\* Directory brute-forcing.

**Vulnerability Summary & Report Card**

**Critical:** 20

**High:** 12

**Moderate:** 2

**Low:** 2

**Informational:** 0

**Network Penetration Test Findings**

**Critical Severity**

* **NPT-000:** Apache 1.3 Vulnerabilities Leading to Remote Code Execution, Directory Traversal, and Denial of Service (CVSS v3 Score: 9.8)
* **NPT-001:** Drupal 7.54 - Remote Code Execution via "Drupalgeddon2" (CVE-2018-7600) (CVSS v3 Score: 9.8)
* **NPT-002:** Samba 2.2.1a "Trans2open" Exploit Leading to Remote Code Execution (CVSS v3 Score: 9.8)
* **NPT-003:** SQL Injection in Login Page Leading to Privilege Escalation (CVSS v3 Score: 9.8)
* **NPT-004:** Command Injection via Ping Function (CVSS v3 Score: 9.8)
* **NPT-005:** SQL Injection Leading to Unauthorized Administrative Access (CVSS v3 Score: 9.8)
* **NPT-006:** Command Injection via Admin Console Ping Function (CVSS v3 Score: 9.8)
* **NPT-007:** Backdoor in vsFTPd 2.3.4 Leading to Remote Code Execution and Root Access (CVSS v3 Score: 9.8)
* **NPT-008:** Remote Code Execution in Samba Service on Port 139 (CVSS v3 Score: 9.8)
* **NPT-009:** Authentication Bypass in VNC Service on Port 5900 (CVSS v3 Score: 9.8)
* **NPT-010:** Misconfigured Samba Service Leading to Root Access (CVSS v3 Score: 9.8)
* **NPT-011:** FTP Misconfiguration - Anonymous Access (CVSS v3 Score: 9.4)
* **NPT-012:** WordPress Plugin Vulnerabilities (CVSS v3 Score: 9.4)
* **NPT-013:** phpMyAdmin Access Unrestricted (CVSS v3 Score: 9.4)
* **NPT-014:** Hardcoded MySQL Credentials (CVSS v3 Score: 9.4)
* **NPT-015:** No Logging or Monitoring (CVSS v3 Score: 9.4)
* **NPT-016:** Insufficient Password Complexity (CVSS v3 Score: 9.4)
* **NPT-017:** Squid Proxy Misconfiguration (3128) (CVSS v3 Score: 9.3)
* **NPT-018:** MySQL Service Exposed on Public Network (CVSS v3 Score: 9.1)
* **NPT-019:** HTTP Access without HTTPS (CVSS v3 Score: 9.1)

**High Severity**

* **NPT-020:** Privilege Escalation via User "Sarah" (CVSS v3 Score: 8.7)
* **NPT-021:** SSH Misconfiguration (CVSS v3 Score: 8.3)
* **NPT-022:** Exposed Sensitive Directories (phpMyAdmin, etc.) (CVSS v3 Score: 8.3)
* **NPT-023:** Weak or Default Credentials on Web-Based Login Page (CVSS v3 Score: 8.2)
* **NPT-024:** Exposure of Sensitive Information via settings.php (CVSS v3 Score: 8.1)
* **NPT-025:** Privilege Escalation via "Dirty COW" Exploit on Ubuntu 14.04.5 (CVE-2016-5195) (CVSS v3 Score: 7.8)
* **NPT-026:** Privilege Escalation via Local Kernel Exploit (CVSS v3 Score: 7.8)
* **NPT-027:** Limited Shell (lshell) Bypass via Python Command Execution (CVSS v3 Score: 7.5)
* **NPT-028:** Outdated ProFTPD Service Vulnerable to Credential Retrieval (CVSS v3 Score: 7.5)
* **NPT-029:** Weak File Permissions (CVSS v3 Score: 7.3)
* **NPT-030:** Open Ports - HTTP (Port 80) (CVSS v3 Score: 7.3)
* **NPT-031:** Use of Weak Password for System User Account (CVSS v3 Score: 7.1)

**Moderate Severity**

* **NPT-032:** No Brute-force Protection on WordPress Login (CVSS v3 Score: 5.3)
* **NPT-033:** FTP Unencrypted Data Transmission (CVSS v3 Score: 5.3)

**Low Severity**

* **NPT-034:** SSH User Enumeration (CVSS v3 Score: 3.7)
* **NPT-035:** Default Configuration for MySQL (CVSS v3 Score: 3.5)

**Technical Findings**

**NPT-000: Apache 1.3 Vulnerabilities Leading to Remote Code Execution, Directory Traversal, and Denial of Service**

**Description:** The Apache 1.3 web server, due to its age and the lack of modern security practices, presents multiple vulnerabilities that can be exploited by attackers. Key issues include directory traversal attacks that allow unauthorized access to sensitive files, remote code execution vulnerabilities that could lead to complete server compromise, and denial of service (DoS) risks from improper request handling. The outdated authentication mechanisms further exacerbate the risk, allowing potential bypasses of security measures. Exploitation of these vulnerabilities can result in significant data breaches, system compromises, and service disruptions, underscoring the critical need for timely updates and robust security configurations.

**Recommended Remediation:**

* Upgrade Apache to a supported version (2.x or later) that receives security updates and has improved security features.
* Harden the Apache configuration by disabling unnecessary modules, restricting access to sensitive files, and implementing security headers.
* Regularly monitor server logs for suspicious activities and perform routine security audits to identify and address potential vulnerabilities.

**System:** Kioptrix Level 1

**References:** <https://www.exploit-db.com/exploits/47080>

**NPT-001: Drupal 7.54 - Remote Code Execution via "Drupalgeddon2" (CVE-2018-7600)**

**Description:** The website, on port 1898, is running Drupal version 7.54, which is affected by the "Drupalgeddon2" vulnerability (CVE-2018-7600). This vulnerability allows an attacker to execute arbitrary code on the server without needing any authentication. Through this exploit, an attacker can gain access to the system as a service user with minimal privileges. The vulnerability stems from improper input validation within the Drupal core, enabling remote code execution (RCE) when exploited. Immediate patching and upgrading are essential to prevent unauthorized system access and maintain the security of the platform.

**Recommended Remediation:**

* Upgrade the Drupal installation to the latest stable version immediately.
* Ensure that security patches are applied regularly to address known vulnerabilities.
* Consider implementing Web Application Firewalls (WAF) to provide an additional layer of protection against known exploits.

**System:** Lampiao

**Tools Used:** Metasploit Framework

**References:** <https://nvd.nist.gov/vuln/detail/CVE-2018-7600>

**NPT-002: Samba 2.2.1a "Trans2open" Exploit Leading to Remote Code Execution**

**Description:** The "Trans2open" vulnerability in Samba version 2.2.1a allows an attacker to exploit a buffer overflow in the SMB (Server Message Block) protocol. Using this exploit, attackers can gain unauthorized root access to the system by sending specially crafted requests to the Samba service. In this case, the attacker is able to execute a reverse shell and achieve root privileges, providing full control over the target machine. This vulnerability poses significant risks as it allows an attacker to manipulate system files, escalate privileges, and potentially use the compromised system as a platform for further attacks.

**Recommended Remediation:**

* Upgrade Samba to a version that is not vulnerable to the "Trans2open" exploit, preferably above version 3.0, where this vulnerability has been patched.
* Minimize the attack surface by disabling unnecessary services and restricting SMB access to trusted users.
* Implement network segmentation to protect vulnerable services and apply firewall rules to limit exposure of SMB services to internal networks only.

**System:** Kioptrix Level 1

**Tools Used:** Metasploit Framework

**References:** <https://nvd.nist.gov/vuln/detail/CVE-2018-7600>

**NPT-003: SQL Injection in Login Page Leading to Privilege Escalation**

**Description:** This vulnerability arises when user inputs are improperly sanitized, allowing attackers to inject and execute arbitrary SQL commands within the database. By leveraging the payload 'OR 1=1-- -', an attacker is able to bypass the login authentication mechanism and gain unauthorized access to the user panels of system accounts, such as John and Robert. Furthermore, these panels expose sensitive information, including plaintext passwords, which can lead to full system compromise. This highlights a severe lack of input validation and data security measures within the web application, posing a critical risk to the confidentiality, integrity, and availability of the system.

**Recommended Remediation:**

* **Input Validation:** Implement parameterized queries (prepared statements) to prevent malicious input from altering the structure of the SQL query.
* **Error Handling:** Ensure that detailed error messages are not exposed to users, which could provide hints for SQL injection attacks.
* **Use Web Application Firewall (WAF):** Use WAF to detect and block SQL injection attempts.
* **Least Privilege:** Ensure the database accounts used by the web application have the minimum required privileges.

**System:** Kioptrix Level 4

**References:** <https://owasp.org/www-community/attacks/SQL_Injection>

**NPT-004: Command Injection via Ping Function**

**Description:** The ping functionality within the web application allows user input to be directly passed to the system shell. By using the semicolon (;), an attacker can concatenate additional commands, leading to command injection. This type of vulnerability permits attackers to execute arbitrary system commands, gaining unauthorized access to sensitive files, executing code, or even obtaining full control over the server. Preventing command injection requires strict input validation and avoiding direct use of user inputs in shell commands.

**Recommended Remediation:**

* Validate and sanitize all user inputs to ensure they do not contain any special characters (such as ;, |, &) that could be used to inject commands.
* Implement input validation to only allow safe inputs for the ping function (e.g., limiting input to IP addresses).
* Consider using a web application firewall (WAF) to detect and block suspicious activity.

**System:** Kioptrix Level 2

**References:** <https://owasp.org/www-community/attacks/Command_Injection>

**NPT-005: SQL Injection Leading to Unauthorized Administrative Access**

**Description:** The login page of the web application was found to be vulnerable to SQL injection, allowing an attacker to bypass authentication and gain unauthorized access to the administrator console. By crafting a specific SQL payload, the attacker tricked the system into accepting invalid login credentials, exploiting the query's logical structure to always evaluate as true. Once authenticated, the attacker could perform administrative actions, including further attacks such as command injection. SQL injection vulnerabilities occur when user inputs are not properly sanitized and allow manipulation of SQL queries, potentially leading to significant data breaches, system manipulation, or unauthorized access.

**Recommended Remediation:**

* Implement parameterized queries (prepared statements) to prevent SQL injection attacks by ensuring user input is treated as data rather than executable code.
* Validate and sanitize all user inputs, especially those involving login mechanisms.
* Consider using web application firewalls (WAFs) to detect and block SQL injection attempts and ensure regular security assessments are conducted to identify and address any vulnerabilities.

**System:** Kioptrix Level 2

**References:** <https://owasp.org/www-community/attacks/SQL_Injection>

**NPT-006: Command Injection via Admin Console Ping Function**

**Description:** The admin console's ping function allowed attackers to inject additional commands by exploiting command injection vulnerabilities. By crafting specific input, the attacker could append shell commands, leading to the execution of arbitrary code on the server. This vulnerability enabled the attacker to establish a reverse shell as the apache user, granting remote command execution capabilities. Command injection vulnerabilities are critical because they allow attackers to bypass security measures, potentially leading to full system compromise. To prevent such attacks, it's essential to sanitize user inputs and ensure no direct access to system commands is permitted through web interfaces.

**Final Recommendations**

To mitigate the identified vulnerabilities and strengthen the security posture of the tested systems, we recommend the following actions:  
- \*\*Upgrade Outdated Software:\*\* Apply patches to outdated services like Apache, Samba, and WordPress.  
- \*\*Strengthen Authentication:\*\* Implement multi-factor authentication (MFA) and enforce strong password policies.  
- \*\*Harden System Configurations:\*\* Disable unused services, configure firewalls, and apply the principle of least privilege.  
- \*\*Regular Monitoring and Audits:\*\* Set up logging and monitoring systems (e.g., Splunk, ELK Stack) to detect malicious activity.

**References**

1. OWASP Top 10 – https://owasp.org/www-project-top-ten/  
2. CVE Details – https://cve.mitre.org/  
3. NIST SP800-53 – https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final  
4. Exploit-DB – https://www.exploit-db.com/