Ethical Hacking & Penetration Testing Lab Report

Name: Anita Kraus
Course: Ethical Hacking
Assignment: DVWA & Metasploit Exploitation Lab

1. Introduction

This document details the setup of a virtual penetration testing lab and the subsequent exploitation exercises performed against known vulnerable systems. The primary objective was to gain practical, hands-on experience with reconnaissance, vulnerability scanning, exploitation of web application and service vulnerabilities, and basic post-exploitation techniques using industry-standard tools within a controlled, isolated environment. The exercises focused on the Damn Vulnerable Web Application (DVWA) and the UnrealIRCd service, both hosted on a Metasploitable 2 virtual machine.

2. Lab Environment Setup

A virtual lab was constructed using Oracle VirtualBox, comprising the following components:

- **Attacker Machine:** Kali Linux (2024.4 virtualbox-amd64) - A Debian-based Linux distribution pre-packaged with numerous penetration testing and digital forensics tools.
- **Target Machine:** Metasploitable 2 - An intentionally vulnerable Linux virtual machine designed for security training and testing purposes.
- **Virtual Network:** Oracle VirtualBox 'NAT Network' named 'HackingNetwork', configured with the IP range 10.0.2.0/24. This allowed isolated network communication between the attacker and target VMs.

IP addresses were confirmed using standard Linux networking commands:

- Kali Linux (Attacker): 10.0.2.5 (ip addr command)
- Metasploitable 2 (Target): 10.0.2.4 (ifconfig command)

3. DVWA Vulnerability Exploitation

This section details the exploitation of vulnerabilities within the Damn Vulnerable Web Application (DVWA) running on Metasploitable 2 (10.0.2.4). All tests were conducted from the Kali Linux VM with the DVWA security level configured to 'Low'. Tools used included Nmap, Firefox, and Burp Suite.

3.1. Step 1: Nmap Scan & DVWA Access

An Nmap TCP SYN scan with service version detection (nmap -sS -sV 10.0.2.4) was performed

against the target. The scan identified Apache httpd 2.2.8 running on port 80 (HTTP). DVWA was accessed via Firefox at http://10.0.2.4/dvwa, and successfully logged into using default credentials (Username: admin, Password: password).

3.2. Step 2: Set Security Level to Low

The DVWA security level was explicitly set to 'Low' via the 'DVWA Security' page in the application's sidebar menu to facilitate demonstration of basic vulnerabilities.

3.3. Step 3: Command Injection Exploit

Navigated to the 'Command Injection' vulnerability page. Inputting 127.0.0.1; whoami into the IP address field resulted in the execution of the whoami command on the server alongside the intended ping command. The output www-data confirmed successful command injection executed under the web server's user context, demonstrating insufficient input sanitization.

3.4. Step 4: SQL Injection Exploit

Navigated to the 'SQL Injection' page. Entering the payload 1' OR '1'='1 into the 'User ID' field manipulated the backend SQL query. This condition always evaluates to true, causing the application to bypass the intended user lookup logic and return all entries from the users table, demonstrating a classic SQL injection vulnerability.

3.5. Step 5: File Upload Exploit (Remote Code Execution)

A simple PHP web shell was created with the content <?php echo system($_GET['cmd']);?> and saved as shell.php. This file was uploaded via the 'File Upload' page. Due to the lack of file type or content validation at the 'Low' security level, the upload was successful. Remote Code Execution (RCE) was achieved by navigating directly to the uploaded shell's URL (http://10.0.2.4/dvwa/hackable/uploads/shell.php) and appending a command via the 'cmd' GET parameter (e.g., ?cmd=whoami). The server executed the command, returning www-data.

3.6. Step 6: Stored XSS (Cross-Site Scripting) Exploit

Navigated to the 'Stored XSS' page. The JavaScript payload <script>alert('Hacked');</script> was entered into the 'Name' and 'Message' fields and submitted. The application stored this input without proper sanitization. Upon revisiting the page, the stored script executed in the browser, triggering an alert box displaying "Hacked", demonstrating a persistent XSS vulnerability.

3.7. Step 7: Configure Burp Suite & Intercept Traffic

Burp Suite was launched on the Kali machine, and its proxy listener was verified to be active on 127.0.0.1:8080. Firefox's network settings were manually configured to route HTTP and HTTPS traffic through this proxy. Subsequent web requests made via Firefox (e.g., browsing, logging into DVWA) were successfully intercepted and logged in Burp Suite's 'Proxy' -> 'HTTP history' tab, demonstrating traffic monitoring capabilities.

3.8. DVWA Exploitation Conclusion

This exercise successfully demonstrated the identification and exploitation of critical web vulnerabilities within DVWA ('Low' security): Command Injection, SQL Injection, Unrestricted File Upload leading to RCE, and Stored XSS. These vulnerabilities primarily stem from a lack of fundamental security controls like input validation and output encoding. The exercise reinforced the utility of Nmap for reconnaissance and Burp Suite for web traffic analysis and manipulation during penetration tests.

4. UnrealIRCd Backdoor Exploitation

4.1. Objective

To identify and exploit the known backdoor vulnerability present in the version of UnrealIRCd running on Metasploitable 2, gain privileged access, and perform basic post-exploitation.

4.2. Step 1: Nmap Scan (Service Identification)

The previous Nmap scan (nmap -sV 10.0.2.4) had already identified UnrealIRCd running on port 6667, a version known to contain a backdoor vulnerability.

4.3. Step 2: Launch Metasploit & Configure Exploit

The Metasploit Framework console (msfconsole) was launched on Kali Linux. The appropriate exploit module was selected and configured:

```
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 10.0.2.4
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
```

Executing run successfully triggered the backdoor and opened a command shell session on the target machine (10.0.2.4).

4.4. Step 3: Confirm Shell Access & Post-Exploitation

Root access was confirmed by executing whoami within the Metasploit shell, which returned root. Basic post-exploitation was performed by exfiltrating the contents of the password files:

```
  # Confirm root access
whoami
# Exfiltrate password files
cat /etc/passwd
cat /etc/shadow
```

4.5. UnrealIRCd Exploitation Conclusion

The known backdoor in the vulnerable UnrealIRCd service was successfully exploited using the

Metasploit Framework, resulting in root-level access to the Metasploitable 2 system. Post-exploitation involved accessing and displaying sensitive system files (/etc/passwd, /etc/shadow). This demonstrates the critical risk associated with running unpatched or known vulnerable network services.

5. Overall Conclusion & Key Learnings

This comprehensive lab exercise provided valuable hands-on experience in:

- **Virtual Lab Environment Setup:** Configuring VMs and virtual networks using Oracle VirtualBox.
- **Reconnaissance:** Utilizing Nmap for port scanning and service/version identification.
- **Vulnerability Identification:** Recognizing common web application flaws (Injection, XSS, File Upload) and known service vulnerabilities (backdoors).
- **Exploitation Techniques:** Applying practical methods to exploit identified vulnerabilities using manual techniques (DVWA) and automated frameworks (Metasploit).
- **Tool Proficiency:** Gaining familiarity with Kali Linux, Metasploit Framework, Nmap, Burp Suite, and standard Linux commands.
- **Web Application Security Concepts:** Understanding the impact of insufficient input validation, output encoding, and insecure file handling.
- **Post-Exploitation Basics:** Achieving privileged access and performing simple data exfiltration.