Task 4: Vulnerability assessment Report (Walkthrough)

Submitted by Athira Biju

1. Introduction

This report presents the findings of a vulnerability assessment and penetration test conducted on the provided virtual machine (ERULNX16.ova). The assessment's goal was to simulate a real-world attack to identify security weaknesses, evaluate their potential impact, and provide clear recommendations for remediation.

2. Methodology

The assessment followed standard penetration testing phases:

2.1 Setup and Discovery

- The target VM was configured on the same network as the Kali Linux assessment machine.
- The target VM's IP address was identified through a network scan (e.g., sudo nmap -SS -O <network-address>/24), locating the target IP (e.g., 10.0.2.9 or 192.168.56.101).

2.2 Enumeration and Scanning

A comprehensive port and service scan (nmap -sC -sV <target-ip>) was conducted, revealing several open ports and services:

Port/Protocol	Service/Version
21/tcp	**ftp** (ProFTPD 1.3.5)
22/tcp	**ssh** (OpenSSH 6.6.1p1 Ubuntu)
80/tcp	**http** (Apache httpd 2.4.7)
445/tcp	**netbios-ssn** (Samba smbd 4.3.11-Ubuntu)
631/tcp	**ipp** (CUPS 1.7)
3306/tcp	**mysql 8080/tcp**
	http (Jetty 8.1.7)

Further enumeration included checking the website manually, using **gobuster** for hidden directories, attempting **anonymous login** on FTP/SMB, and searching for known **CVEs** for all exposed services.

2.3 Exploitation and Privilege Escalation

1. **Initial Access:** Exploited the vulnerable **ProFTPD 1.3.5** service using the **mod_copy** vulnerability (CVE-2015-3306) to gain a reverse shell (e.g., using Metasploit with a Perl payload).
2. **Web Vulnerability:** Identified and successfully exploited a **Cross-Site Scripting (XSS)** vulnerability on the web application's chat page using the payload <script>alert(123)</script>.
3. **Privilege Escalation:** After obtaining a low-privileged shell (e.g., as the www- data user), system checks like sudo -l were performed to find misconfigurations or exploits to escalate privileges to the **root** user.
4. **Proof of Compromise:** Root access was validated by accessing sensitive files (e.g., /etc/shadow) and capturing the final flag (/root/flag.tx

3.Findings and Risk Assessment

The assessment demonstrated that the VM was severely vulnerable, allowing for complete system compromise.

Vulnerability / Service	Risk Level	Description	CVSS Score
ProFTPD 1.3.5	CRITICAL	Vulnerable to **Remote Code Execution (RCE)** via the mod_copy module (CVE-2015-3306).	**G.8**
Apache httpd 2.4.7	High	Outdated version with multiple known vulnerabilities, including successful **Cross-Site Scripting (XSS)**.	**6.1**
Samba smbd 4.3.11	High	Older version with known RCE issues and **message signing disabled** (enabling Man-in-the-Middle attacks).	**8.1**
Jetty 8.1.7	Medium	Outdated version (2012) with known information disclosure vulnerabilities.	**6.8**
CUPS 1.7	Medium	Exposes the IPP service with risky methods (e.g., **PUT**) that could be abused for file uploads.	**6.5**
MySQL (unauthorized)	Medium	MySQL port open to the external network, increasing the overall attack surface.	**6.5**
OpenSSH 6.6.1p1	Medium	Outdated; known vulnerabilities related to information disclosure and weak ciphers.	**5.G**

4.Recommendations

Immediate action is required to patch these critical security weaknesses:

1. **Critical Patch Management: Immediately update or disable the ProFTPD service** to the latest secure version to mitigate the Remote Code Execution vulnerability.
2. **Secure Web Application:** Implement robust **input validation and output sanitization/encoding** across the web application to prevent Cross-Site Scripting (XSS) attacks. Upgrade to the latest stable Apache release.
3. **Service Upgrades and Hardening:**
    - **Samba:** Enable message signing, patch the service, and restrict its network exposure.
    - **OpenSSH/CUPS/Jetty:** Upgrade all outdated services to supported versions and implement secure configurations (e.g., disable risky methods, restrict network access).
4. **Network Firewalling:** Restrict the **MySQL** service to listen only on localhost (internal connections) and use network firewalls to block all unauthorized external access to ports like 3306 and 8080.

These measures are essential for preventing complete system compromise and maintaining a secure environment.