A Security Checkup: Your VM's Vulnerability Story

Prepared by Sreehari Vinod

This report summarizes a successful security investigation into your Virtual Machine (ERULNX16.ova).

1. Introduction: The Mission

This document tells the story of our penetration test. We approached your virtual machine just like a motivated threat actor would: to find every hidden weakness, exploit it, and gain complete control.

The good news? We succeeded, meaning the flaws are real. The great news? We've given you the clear, actionable plan to fix every single one of them. Your system is exposed, but it's completely fixable.

2. Methodology: Our Digital Investigation

Our process was a four-step digital stakeout:

2.1 First Contact & Fingerprinting

- **Discovery:** We found the target VM's address on the network (for example, 10.0.2.9).
- **Scanning**: We ran a comprehensive scan to check every "door and window," which immediately revealed several open services that should not be visible, including FTP on port 21, the main web server on 80, and a secondary Jetty server on 8080.

2.2 The Breach & The Climb

- **Initial Access:** The single biggest vulnerability was the ProFTPD 1.3.5 service. We used a known, critical flaw called mod_copy (CVE-2015-3306) to bypass security and instantly execute code, giving us a shell on the system. We also found a major flaw in your web app, successfully proving a Cross-Site Scripting (XSS) attack was possible on the chat page.
- **The Finish Line:** Once inside with low-level access (as the www-data user), we searched the system for any misconfiguration that would let us upgrade our privileges. We found the path and successfully became the root user, confirming a total system compromise.

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.txt).

3. The Findings: Critical Risks Identified

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

- **CRITICAL Flaw (ProFTPD 1.3.5):** This is the most severe vulnerability. The FTP service is exposed to a well-known exploit (CVE-2015-3306) that allows **Remote Code Execution (RCE)**, giving an attacker complete control over the system.
- **High Risk (Apache 2.4.7 & Samba 4.3.11):**
    - The Apache web server is outdated and has a proven **Cross-Site Scripting (XSS)** vulnerability.
    - The Samba file-sharing service has **message signing disabled**, which leaves it open to Man-in-the-Middle (MITM) attacks and is an older version with known RCE issues.
- **Medium Risk (Exposure & Outdated Services):**
    - The **MySQL** port is unnecessarily open to the external network, significantly increasing the attack surface.
    - Other services, including **Jetty 8.1.7** and **OpenSSH 6.6.1p1**, are outdated and vulnerable to various information disclosure issues.

4. Your Action Plan: How to Secure Everything

Immediate action is required to patch these critical security weaknesses:

1. **The Quick Fix (Critical Patching): Stop using ProFTPD 1.3.5 immediately.** You must update this service to the latest secure version or disable it entirely if it is not absolutely essential to your operations.
2. **Guard the Website (Input Validation**): Update to the latest stable Apache release. More importantly, you must implement strong input validation and sanitization across the entire web app to prevent the XSS attacks we demonstrated.
3. **Close the Digital Doors (Firewalling):**
   - Your **MySQL** database should only ever listen to localhost (internal connections).
   - Use network firewalls to block all unauthorized external access to ports like 3306 and 8080.
4. **Harden the Core (Upgrades):** Upgrade all outdated services, including OpenSSH, CUPS, and Jetty, to supported and modern versions. For Samba, you must enable message signing to shut down the Man-in-the-Middle risk.

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