**Wireshark**

Samuel Folaranmi

Professor Ingrid Gaviria

ITT - 307 - 0500

June 6, 2024

**Network and Vulnerability Scanning with Wireshark Monitoring**

**Overview**

The purpose of this assignment is to conduct a network and vulnerability scan using Nmap and OpenVAS on the virtual network, while simultaneously capturing network traffic with Wireshark. This approach helps in understanding the interaction between scanning tools and the target systems, and how network traffic is generated and analyzed during security assessments (Lyon, 2023).

**Start Wireshark on Kali Linux**



Start capturing traffic



OS fingerprinting scan using Nmap



service version detection scan using Nmap



Results



In this assignment, we successfully performed a network and vulnerability scan using Nmap and OpenVAS while capturing the network traffic with Wireshark. The Nmap scans accurately identified the OS and services running on the target Metasploitable VM, and the OpenVAS scan identified multiple critical vulnerabilities. Wireshark effectively captured the network traffic generated during these scans, allowing us to analyze the interactions and the type of data transmitted during the scanning processes (Lyon, 2023).

**Results of Wireshark Analysis:**

* **Nmap Traffic:**
	+ Multiple SYN packets sent to identify open ports.
	+ Detailed service detection probes sent to determine versions of services.
* **OpenVAS Traffic:**
	+ Extensive network traffic as OpenVAS probes for vulnerabilities and gathers detailed information about the target systems.

**Recommendations:**

Based on the identified vulnerabilities, the following recommendations are provided:

1. **Regular Updates and Patching:**
	* Keep all software and systems up to date with the latest security patches to mitigate known vulnerabilities.
2. **Secure Configurations:**
	* Harden configurations of services to disable insecure and unnecessary features.
3. **Network Segmentation and Firewalls:**
	* Implement network segmentation and use firewalls to limit access to sensitive services and reduce the attack surface.
4. **Continuous Monitoring and Scanning:**
	* Conduct regular security scans and continuous monitoring to identify and address new vulnerabilities promptly (Greenbone Networks, 2023).

**Difference Between Vulnerability Scanning and Penetration Testing**

**Vulnerability Scanning:**

- Definition: Automated process that identifies potential vulnerabilities in systems, networks, and applications.

- Purpose: To detect known vulnerabilities, misconfigurations, and outdated software.

- Tools: Common tools include OpenVAS, Nessus, and QualysGuard.

- Output: Generates a list of vulnerabilities with severity levels, descriptions, and remediation suggestions.

- Frequency: Often performed regularly as part of routine security maintenance.

- Depth: Generally broad but not deep; it identifies vulnerabilities but does not exploit them.

- Example: Using OpenVAS to scan a network for known vulnerabilities and producing a report with the findings.

**Penetration Testing:**

- Definition: Simulated cyber-attack performed by security professionals to exploit vulnerabilities.

- Purpose: To identify security weaknesses by actively exploiting them, demonstrating what an attacker could do.

- Tools: May include manual techniques, custom scripts, and tools like Metasploit, Burp Suite, and Wireshark.

- Output: Provides a detailed report on exploited vulnerabilities, the method of exploitation, and recommendations for remediation.

- Frequency: Performed less frequently, often annually or bi-annually, or when significant changes occur in the network.

- Depth: Deep and thorough; it involves exploiting vulnerabilities to assess the impact.

- Example: Conducting a penetration test to exploit a vulnerable service on a server, demonstrating how an attacker could gain unauthorized access.

 **Using Wireshark Data to Identify Differences Between Scans**

Wireshark captures network traffic, which can be analyzed to differentiate between vulnerability scans and penetration tests based on their network behavior.

 **Characteristics of Vulnerability Scans in Wireshark:**

- Pattern of Traffic: High volume of automated requests probing various ports and services.

- Types of Packets: Numerous SYN packets to multiple ports, followed by service-specific probes.

- Destination Ports: A wide range of ports as the scanner identifies services running on the network.

- Content: Packets may include version banners and service information requests.

- Frequency: Continuous and repetitive probing.

 **Characteristics of Penetration Tests in Wireshark:**

- Pattern of Traffic: More targeted and strategic probing based on identified vulnerabilities.

- Types of Packets: Exploit packets, custom payloads, and abnormal sequences indicating exploitation attempts.

- Destination Ports: Specific ports identified as vulnerable during the initial reconnaissance.

- Content: Packets may include exploit payloads, shell commands, and unusual traffic patterns.

- Frequency: Less frequent but more varied in type, aiming to exploit vulnerabilities.

 **Example Analysis Using Wireshark:**

1. Vulnerability Scan Traffic:

 - Numerous SYN packets to a range of ports.

 - Follow-up requests probing specific services (e.g., HTTP, FTP, SSH).

 - Patterns indicating systematic checks for service versions and configurations.

2. Penetration Test Traffic:

 - SYN packets to known vulnerable ports.

 - Exploit attempts with payloads in the data section of the packets.

 - Communication attempts that suggest command-and-control activities.

 **Using Wireshark Data for Forensics Investigation:**

- Identifying Patterns: Analyzing traffic patterns can help differentiate between normal, scanning, and exploitative behavior.

- Tracking Attacker Activity: Wireshark logs can show the sequence of actions an attacker takes, including reconnaissance, exploitation, and post-exploitation activities.

- Correlating Events: By correlating suspicious network traffic with system logs, security analysts can track the attacker's movements and methods.

- Timeline Reconstruction: Wireshark's timestamps can be used to create a timeline of the attack, detailing when each step occurred.

- Incident Response: Detailed packet captures can provide evidence for incident response teams to understand the scope and impact of an attack.

**References:**

Greenbone Networks. (2023). *Greenbone Vulnerability Management*. Retrieved from https://www.greenbone.net/en/vulnerability-management/

Lyon, G. F. (2023). *Nmap Network Scanning*. Insecure.Org. Retrieved from https://nmap.org/book/man-version-detection.html