MODULE 08 SNIFFING LAB REPORT

Lara Alofi

VULNERABILITY ANALYSIS AND DEFENSE
Course Code: T44-17520

Lab Session Identifiers

https://labclient.labondemand.com/LabClient/1b0961e2-f115-495b-b6d8-0b95128ee84a

Username on EC-Council System

1. 2110886@uj.edu.sa

Lab 01:

Task 1: Perform MAC Flooding using macof

1. **Introduction to MAC Flooding**: Understand the concept of MAC flooding, a technique to compromise network switches by overwhelming their MAC address tables.
2. **Using macof Tool**: Learn about the **macof** tool, which floods the network with random MAC addresses, causing switches to behave like hubs and facilitating packet sniffing.
3. **Launching the Attack**: Execute the attack by flooding the network with random MAC addresses using the **macof** tool.
4. **Observing the Effect**: Use Wireshark to observe the flooding effect, capturing packets with random MAC addresses.
5. **Understanding the Impact**: Recognize the impact of MAC flooding on network security, as it can lead to unauthorized access and packet interception.

Key Learning: MAC flooding can compromise network security by forcing switches to behave like hubs, making it easier for attackers to intercept and manipulate network traffic.

Task 2: Perform a DHCP Starvation Attack using Yersinia

1. **Understanding DHCP Starvation Attack**: Learn about DHCP starvation attack, which exhausts DHCP IP address pool, causing denial of service.
2. **Introduction to Yersinia**: Explore the **Yersinia** tool, designed to exploit weaknesses in network protocols like DHCP.
3. **Launching the Attack**: Execute DHCP starvation attack using Yersinia to flood the DHCP server with DHCP requests, preventing valid users from obtaining IP addresses.
4. **Observing the Effect**: Use Wireshark to observe the flood of DHCP requests and the impact on network availability.
5. **Impact and Consequences**: Understand the consequences of DHCP starvation attack, including network downtime and disruption of services.

Key Learning: DHCP starvation attack can disrupt network services by exhausting DHCP IP address pool, preventing valid users from obtaining IP addresses.

Task 3: Perform ARP Poisoning using arpspoof

1. **Understanding ARP Poisoning**: Learn about ARP poisoning attack, intercepting network traffic by spoofing ARP replies.
2. **Introduction to arpspoof**: Explore the **arpspoof** tool, used to redirect traffic through the attacker's machine by poisoning ARP cache.
3. **Executing the Attack**: Perform ARP poisoning attack using arpspoof to intercept and manipulate network traffic between two systems.
4. **Observing the Effect**: Use Wireshark to observe the ARP poisoning effect, capturing spoofed ARP packets.
5. **Recognizing the Vulnerability**: Understand how ARP poisoning exposes the vulnerability of ARP protocol to interception and manipulation.

Key Learning: ARP poisoning allows attackers to intercept and manipulate network traffic by spoofing ARP replies, highlighting the importance of ARP security measures.

Task 4: Perform an Man-in-the-Middle (MITM) Attack using Cain & Abel

1. **Understanding MITM Attack**: Learn about MITM attack, intercepting communication between two parties to capture sensitive information.

2. **Introduction to Cain & Abel**: Explore the **Cain & Abel** tool, used to capture usernames and passwords by sniffing network traffic.
3. **Executing the Attack**: Perform MITM attack using Cain & Abel to capture FTP credentials between two systems.
4. **Observing the Effect**: Use Cain & Abel to observe captured usernames and passwords, highlighting the vulnerability of unencrypted network communication.
5. **Recognizing the Risk**: Understand the risk posed by MITM attacks to sensitive information transmitted over unsecured networks.

Key Learning: MITM attacks can intercept and capture sensitive information transmitted over unencrypted networks, emphasizing the importance of encryption and secure communication protocols.

Task 5: Spoof a MAC Address using TMAC and SMAC

1. **Understanding MAC Spoofing**: Learn about MAC address spoofing, changing the MAC address of a network device to impersonate another device or evade detection.
2. **Introduction to TMAC and SMAC**: Explore the **TMAC** and **SMAC** tools, used to change the MAC address of network adapters.
3. **Executing MAC Spoofing**: Perform MAC spoofing using TMAC and SMAC tools to simulate network attacks while maintaining anonymity.
4. **Observing the Effect**: Observe the changed MAC address and recognize its implications for network security and detection.
5. **Recognizing the Implications**: Understand how MAC spoofing can be used to evade network restrictions and impersonate other devices on the network.

Key Learning: MAC spoofing allows attackers to evade detection and impersonate other devices on the network, highlighting the importance of MAC address security measures.

Task 6: Spoof a MAC Address of Linux Machine using macchanger

1. **Understanding MAC Address Spoofing**: Learn about MAC address spoofing on Linux systems, changing the MAC address of a network interface.
2. **Introduction to macchanger**: Explore the **macchanger** utility, used to change the MAC address of network interfaces on Linux systems.
3. **Executing MAC Spoofing**: Perform MAC spoofing using **macchanger** to change the MAC address of a network interface on a Linux machine.
4. **Observing the Effect**: Observe the changed MAC address and recognize its implications for network anonymity and security.
5. **Understanding the Process**: Understand the process of MAC spoofing on Linux systems and its role in enhancing network anonymity.

Key Learning: MAC spoofing on Linux systems can enhance network anonymity and evade detection, emphasizing the importance of MAC address security measures.

By following these steps and understanding the key learnings from each task, one can gain practical insights into various network attacks and the tools used to execute them, as well as the importance of network security measures to defend against such attacks.

Lab 02:

Task 1: Perform Password Sniffing using Wireshark, along with key learnings:

1. **Introduction to Wireshark:** Understand Wireshark as a network packet analyzer used to capture and analyze network packets in detail.
2. **Setup:** Configure two machines - Windows Server 2019 (host) and Windows 11 (target).
3. **Capturing Packets on Host Machine (Windows Server 2019):**
    - Launch Wireshark.
    - Start capturing packets on the Ethernet interface.
    - Wait for traffic to be generated.
4. **Logging into Target Machine (Windows 11):**
    - Log in to Windows 11 using provided credentials (Admin: Pa$$w0rd).
5. **Browsing on Target Machine:**
    - Open a web browser and navigate to a website (e.g., moviescope.com).
    - Enter credentials (Username: sam, Password: test) and log in.
6. **Analyzing Captured Packets:**
    - Stop packet capture on the host machine (Windows Server 2019).
    - Save captured packets for analysis.
    - Apply a display filter to narrow down to HTTP POST traffic.
    - Search for packets containing the string "pwd" to find potential passwords.
    - View captured username and password details.
7. **Remote Desktop Connection:**
    - Establish a remote desktop connection to the target machine (Windows 11) from the host (Windows Server 2019).
    - Start the Remote Packet Capture Protocol service on the target machine.
8. **Capture Packets Remotely:**
    - Configure Wireshark on the host machine to capture packets remotely from the target machine.
    - Start capturing packets remotely.
9. **Activity on Target Machine:**
    - Log in to the target machine (Windows 11) again.
    - Browse the Internet or perform other activities to generate network traffic.
10. **Stop Capturing Packets:**
    - Stop capturing packets on the host machine (Windows Server 2019) after sufficient activity.

Key Learnings:

- Wireshark can capture and analyze network traffic, including sensitive information like passwords.
- Captured packets may contain plaintext credentials, making them vulnerable to sniffing.
- Remote packet capture allows monitoring of network traffic on remote machines, useful for analyzing network activities and potential security threats.
- Password sniffing demonstrates the importance of encryption and secure communication protocols to protect sensitive information during transmission over networks.

Task2 and 3 doesn't worked.

Lab 03:

Task 1: Detect ARP Poisoning and Promiscuous Mode in a Switch-Based Network, along with key learnings:

1. **Introduction to ARP Poisoning and Promiscuous Mode:**
    - Understand ARP poisoning as a method to manipulate ARP caches and intercept traffic, and promiscuous mode as a mode that allows a device to capture all network traffic.
    - Recognize the potential threats posed by ARP poisoning, including data interception, DoS attacks, and MITM attacks.
2. **Setup:**
    - Configure three machines: Windows Server 2019 (host), Windows 11 (target), and Parrot Security.
    - Install necessary tools such as Cain & Abel, Wireshark, and Nmap on the host machine.
3. **Perform ARP Poisoning with Cain & Abel:**
    - Use Cain & Abel to configure the network adapter and start ARP poisoning between Windows 11 and Parrot Security.
    - Verify ARP poisoning status in Cain & Abel.
4. **Generate Traffic:**
    - Use hping3 on Parrot Security to generate traffic towards Windows 11.
5. **Capture Packets with Wireshark:**
    - Launch Wireshark on the host machine and configure it to capture packets on the appropriate network interface.
    - Observe captured packets to detect ARP poisoning warnings.
6. **Analyze Packet Capture:**
    - Investigate Wireshark's Expert Information to identify duplicate IP address warnings, indicating potential ARP poisoning**.**
7. **Detect Promiscuous Mode with Nmap:**
    - Switch to Windows 11 and use Zenmap to perform a scan and detect promiscuous mode on the host machine (Windows Server 2019).
    - Analyze the scan results to confirm promiscuous mode detection.
8. **Document Findings:**
    - Close all tools and windows and document the acquired information, including ARP poisoning detection and promiscuous mode detection.

Key Learnings:

- ARP poisoning and promiscuous mode are common network security threats that can lead to data interception and manipulation.
- Tools like Cain & Abel, Wireshark, and Nmap can be used to detect and analyze these threats.
- Understanding how to detect and mitigate ARP poisoning and promiscuous mode is crucial for network security professionals and ethical hackers.
- Regular monitoring and analysis of network traffic can help identify and prevent potential security breaches.

Task 2: Detect ARP Poisoning using the Capsa Network Analyzer:

1. **Introduction to Capsa Network Analyzer:**
    - Understand Capsa as a network performance analysis and diagnostics tool used for packet capture and analysis.
    - Recognize its capabilities in detecting ARP poisoning and ARP flooding attacks.
2. **Installation of Capsa Network Analyzer:**
    - Download Capsa Enterprise Trial from the Colasoft website.
    - Install and activate Capsa Enterprise on the host machine (Windows Server 2019).
3. **Setup and Configuration:**
    - Launch Capsa Enterprise and select the network adapter to monitor.
    - Start capturing packets on the selected adapter.
4. **Perform ARP Poisoning with Habu:**
    - Use Habu on Parrot Security to perform ARP poisoning towards Windows 11.
    - Observe the ARP poisoning effects in Capsa Enterprise.
5. **Detect ARP Poisoning with Capsa:**
    - Analyze the captured packets in Capsa Enterprise to detect ARP poisoning warnings.
    - Resolve IP addresses to identify the attacker's IP.
6. **Generate Diagnostic Logs:**
    - Configure Capsa to save diagnostic logs to disk at regular intervals.
    - Analyze the logs to track network behavior and detect anomalies.
7. **Document Findings:**
    - Close all tools and windows, and document the acquired information, including ARP poisoning detection and diagnostic logs.

Key Learnings:

- Capsa Network Analyzer is a powerful tool for detecting network anomalies, including ARP poisoning.
- ARP poisoning can be detected by analyzing ARP request storms and excessive ARP replies.
- Regular monitoring and logging of network traffic are essential for detecting and mitigating ARP poisoning attacks.
- Ethical hackers and network administrators should be familiar with tools like Capsa and Habu for effective network security monitoring and analysis.