Sniffing in a Controlled Environment - Lab Report

Objective:

The purpose of this lab is to demonstrate how credentials transmitted in plaintext over an unencrypted network can be captured using sniffing tools like Wireshark. This exercise simulates a basic real-world attack scenario to highlight the risks of insecure data transmission and the importance of encryption.

Lab Environment Setup:

1. Virtual Machines Used:

- Attacker Machine: Kali Linux (latest version)

- Victim Machine: Windows 10

2. Virtualization Software: VMware Workstation / VirtualBox

3. Network Configuration:

- Network Type: Host-Only / Bridged Adapter

- Both machines configured on the same subnet

- Verified connectivity using ping from both machines

Tools Used:

1. Wireshark: Packet capture and network protocol analyzer tool used to sniff and analyze traffic

2. Browser (on Windows): Used to simulate login to an HTTP test site

3. Test Website: http://testphp.vulnweb.com/login.php (Unencrypted HTTP site for credential testing)

Step-by-Step Process:

Step 1: Launch and Verify VMs

Boot both Kali Linux and Windows 10 virtual machines.
Ensure they are running and connected to the same virtual network.
Verify connectivity by pinging from Kali to Windows and vice versa.

Step 2: Start Wireshark on Kali Linux

Open Wireshark with root privileges.
Identify the active network interface (e.g., eth0 or enp0s3) using 'ip a'.
Select the correct interface in Wireshark.
Apply filter: 'http' to monitor only HTTP traffic.
Click on the 'Start Capturing' button.

Step 3: Simulate Credential Transmission

On the Windows 10 machine, open a web browser.
Navigate to the test site: http://testphp.vulnweb.com/login.php
Enter fake credentials (e.g., (1)username: Mahavir, password: 1a2b3c (2)username: kali, password: 142536).
Click on the login button to transmit data.

Step 4: Capture and Save Packet

Return to Kali Linux with Wireshark running.
Identify the HTTP POST request packet in Wireshark.
Right-click on the packet and choose 'Follow' > 'TCP Stream'.
Extract the plaintext credentials from the stream.
Save the capture file as '.pcapng' for reporting.

Step 5: Analyze and Extract Credentials

In the TCP Stream window, locate the data:
Frame No. 2669
POST /login.php HTTP/1.1
Host: testphp.vulnweb.com
username=Mahavir&password=1a2b3c
This confirms successful sniffing of credentials.

Deliverables:

1. Screenshot of Wireshark Capture:

- Showing HTTP POST request with captured username and password.

2. Short Explanations:

a. Risks of Transmitting Data in Plaintext:

- Plaintext transmission allows attackers to intercept sensitive data.
- Unencrypted protocols like HTTP and FTP expose credentials.
- Leads to identity theft, unauthorized access, and data breaches.

b. Tools Used and Their Functions:

- Wireshark: Sniffs and analyzes live network traffic.
- Kali Linux: Penetration testing platform for attackers.
- Windows 10: Simulates a real user sending insecure credentials.
- Browser + HTTP Site: To create the vulnerable login scenario.

c. How Sniffing Can Be Prevented (Technical Controls):

- Use HTTPS: Encrypts data using SSL/TLS.
- Avoid Plaintext Protocols: Use SFTP/SSH instead of FTP/Telnet.
- Switches instead of Hubs: Prevent traffic broadcast to all systems.
- Network Segmentation: Isolate sensitive systems.
- VPNs: Encrypt traffic over public networks.
- Use IDS/IPS: Detect and block sniffing behavior.

Conclusion:

This lab successfully demonstrated the risk of transmitting sensitive information over unencrypted protocols. Tools like Wireshark can easily capture such data when proper encryption is not in place. It emphasizes the importance of secure protocols (HTTPS) and robust network configurations to protect data integrity and confidentiality.

Note: All testing was conducted in a safe, controlled lab environment strictly for educational purposes.