Experiment 4(b): Sniffing Attack (Password Capture with Wireshark)

Lab Objective

Demonstrate how an attacker can capture clear-text passwords from network traffic using Wireshark in a local LAN environment.

Background

Network Sniffing Overview[1][2]

Network sniffing is the process of intercepting and logging traffic that passes over a digital network. Sniffers (like Wireshark) can capture sensitive data such as credentials if the traffic is unencrypted (e.g., HTTP). This attack is most effective on networks where the attacker and victim share the same LAN, such as public Wi-Fi or switched networks.

Key Concepts:

- Sniffers work in “promiscuous mode”, capturing all packets visible to network interface[1]
- Encrypted protocols (HTTPS, SSH) prevent easy sniffing; plain protocols (HTTP, Telnet, FTP) are vulnerable[2]
- Passwords and data sent in clear text can be read directly from captured packet contents

---

Pre-Lab Requirements

- Two machines: Victim and Attacker
- Both are connected to the same local area network (LAN)
- Attacker machine: Wireshark installed and root privileges
- Victim machine: Browser and access to a test website (HTTP login, not HTTPS)
- Test web application or page for login
- Basic knowledge of network interfaces and packet analysis

Lab Setup

Step 1: Setup the Test Environment

1. Connect both victim and attacker to the same network (Wi-Fi/LAN)
2. Choose a test website that uses HTTP (not HTTPS) with a login page (for safe lab, use a local vulnerable app)

Step 2: Start Wireshark on Attacker

1. Open Wireshark on the attacker’s machine
2. Select the correct network interface (e.g., eth0, wlan0)
3. Set Wireshark to “promiscuous mode” (enabled by default)
4. Click “Start capturing”

Experimental Procedure

Step 3: Generate Login Traffic from Victim

1. On victim machine, open HTTP login page in browser
2. Enter sample username (e.g., “alice”) and password (e.g., “password123”)
3. Click “Login” to submit credentials
4. This action generates HTTP POST request with clear-text credentials

Step 4: Filter and Find Passwords in Wireshark

1. On attacker/Wireshark machine, stop capture after form submission
2. Use the filter: `http.request.method == "POST"` or just `http`
3. Inspect packet “Info” field in the middle panel for POST submission
4. In lower panel, navigate to HTTP payload (look for text like `username=alice&password=password123`)
5. Note the captured username and password in plain text

Example Screenshot

Figure 1: Packet capture in Wireshark showing clear-text HTTP credentials

---

Observations & Results

Action	Attacker Observation
Victim logs in via HTTP	Traffic captured by Wireshark
POST packet filtered	Username and password found in plain text

Table 1: Lab observations: Capturing plaintext credentials using Wireshark

---

Analysis

1. **Why was the password visible to the attacker?**
    - HTTP does not encrypt payload data; credentials travel as plain text.
    - Wireshark, in promiscuous mode, reads all network packets on the LAN segment[1][2].
2. **Why is HTTPS not vulnerable to this attack?**
    - HTTPS encrypts traffic between browser and server.
    - Even if packets are captured, contents appear as unreadable ciphertext.
3. **How can attackers leverage this vulnerability?**
    - Gain unauthorized access to victim accounts.

- Use captured data for further attacks (phishing, session hijacking).

Defense & Mitigation

1. **Always use HTTPS for login pages**
    - Never enter credentials on HTTP sites
2. **Use strong, unique passwords for every account**
3. **Disable unencrypted protocols on networks**
4. **Monitor network for suspicious packet sniffing**
5. **Educate users about safe web practices**

Viva Questions & Answers

Q1: What is network sniffing?
A1: Network sniffing is intercepting and inspecting network traffic, often using tools like Wireshark, to capture packets that may contain sensitive data.

Q2: Why are HTTP logins vulnerable to sniffing attacks?
A2: HTTP transmits data in plain text, so anyone capturing traffic on the network can read usernames and passwords directly.

Q3: Can Wireshark capture passwords over HTTPS?
A3: No, HTTPS encrypts all data between the client and server. Wireshark can capture packets, but the data will be encrypted and unreadable.

Q4: List one way to prevent password sniffing on a network.
A4: Always use HTTPS (SSL/TLS) to encrypt communications, especially login pages.

Conclusion

This lab demonstrated how an attacker with access to the same network segment as a victim can use Wireshark to capture plaintext credentials sent over HTTP. It highlights the critical importance of using encrypted protocols for any sensitive information on the internet.

References

[1] ECCouncil. (2025). Network Packet Capturing and Analysis with Wireshark. https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/wireshark-packet-capturing-analysis/

[2] SEED Security Labs. (2025). Packet Sniffing and Spoofing Lab. https://seedsecuritylabs.org/Labs_20.04/Files/Sniffing_Spoofing/Sniffing_Spoofing.pdf