Network Traffic Analysis with Wireshark and tcpdump

Project Overview

This project demonstrates my ability to capture, analyse, and interpret network traffic using **Wireshark** and **tcpdump** on a local **DVWA (Damn Vulnerable Web Application)** environment. The goal was to perform passive reconnaissance by sniffing network packets, identifying key communication patterns, and extracting sensitive information such as login credentials and session cookies from unencrypted HTTP traffic.

Key Skills Demonstrated

- Local network traffic capture using **tcpdump**
- Packet analysis with **Wireshark**
- HTTP traffic inspection on 127.0.0.1 (localhost)
- Identification of security vulnerabilities in unencrypted communications

---

Project Objectives

1. **Prepare the host for local network traffic capture.**
2. **Capture and save HTTP traffic between the browser and DVWA.**
3. **Analyse the captured traffic using Wireshark to extract meaningful data.**

Tools Used

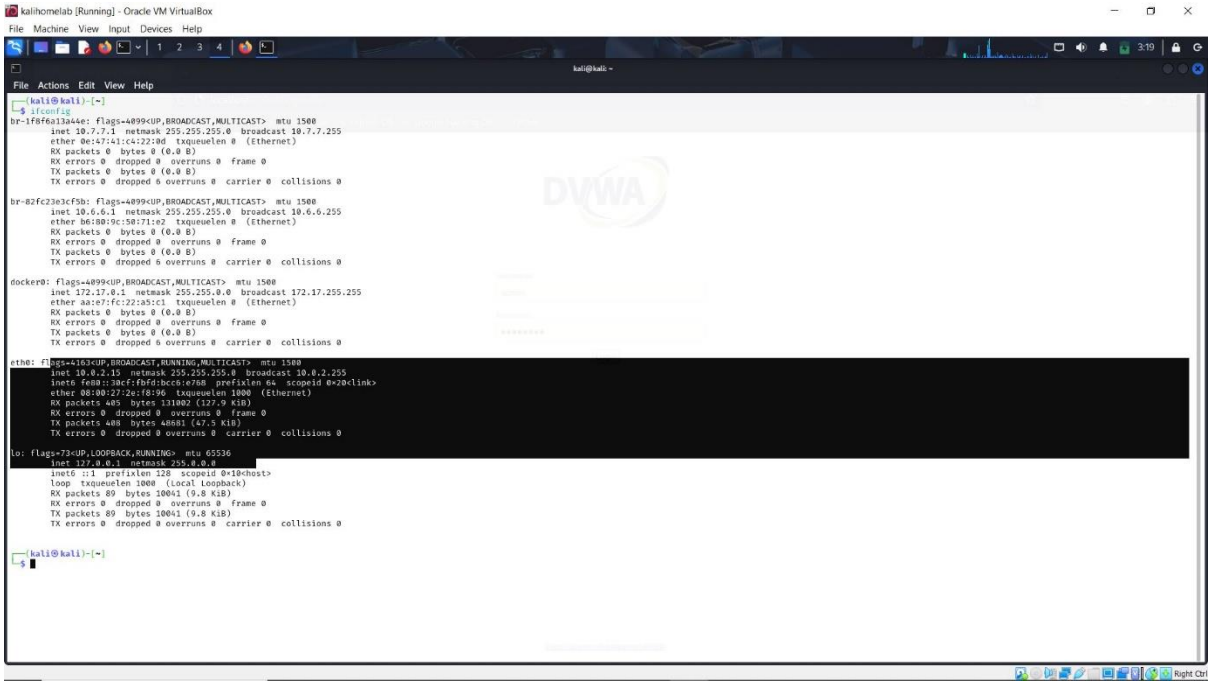
- **Kali Linux VM** (for packet capture and analysis)
- **tcpdump** (command-line packet sniffer)
- **Wireshark** (GUI-based network protocol analyser)
- **DVWA (Damn Vulnerable Web Application)** running on http://127.0.0.1

Methodology

Part 1: Preparing the Host for Traffic Capture

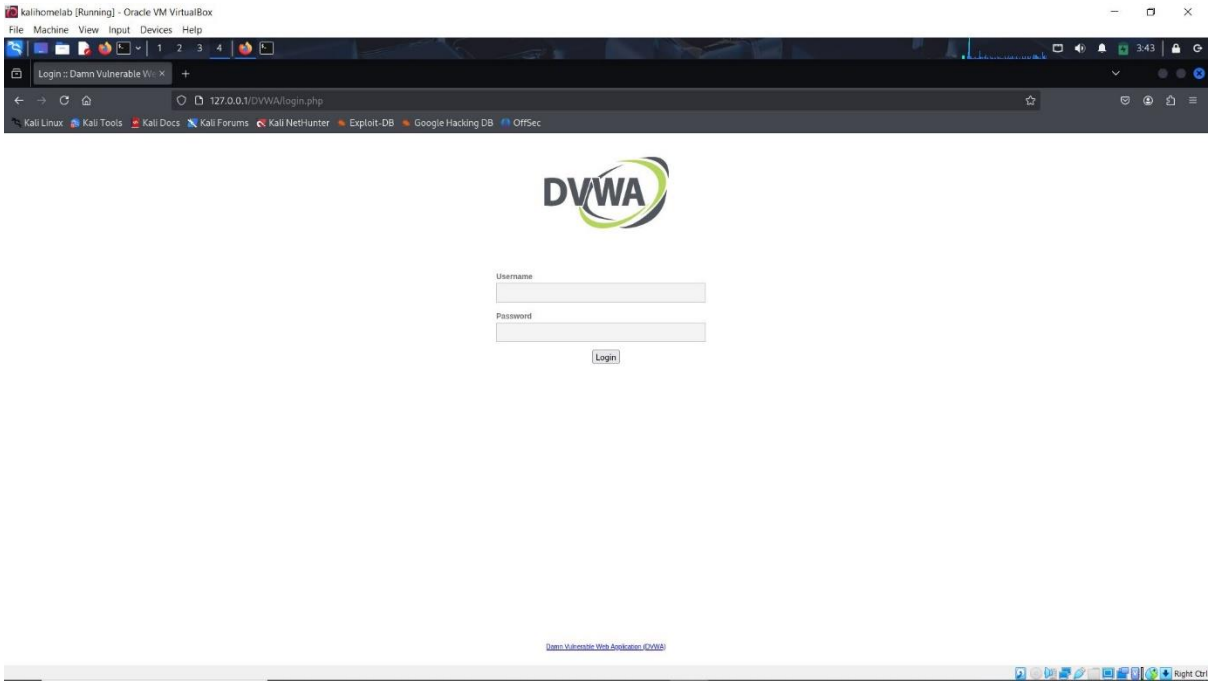
1. **Logged into Kali Linux**

2. **Determined network interface details** (lo for loopback/localhost) using: **ifconfig**



IP address: 10.0.2.15, Mac address 08:00:27:2e:f8:96 is the source address for packets sent from my Kali Linux machine.

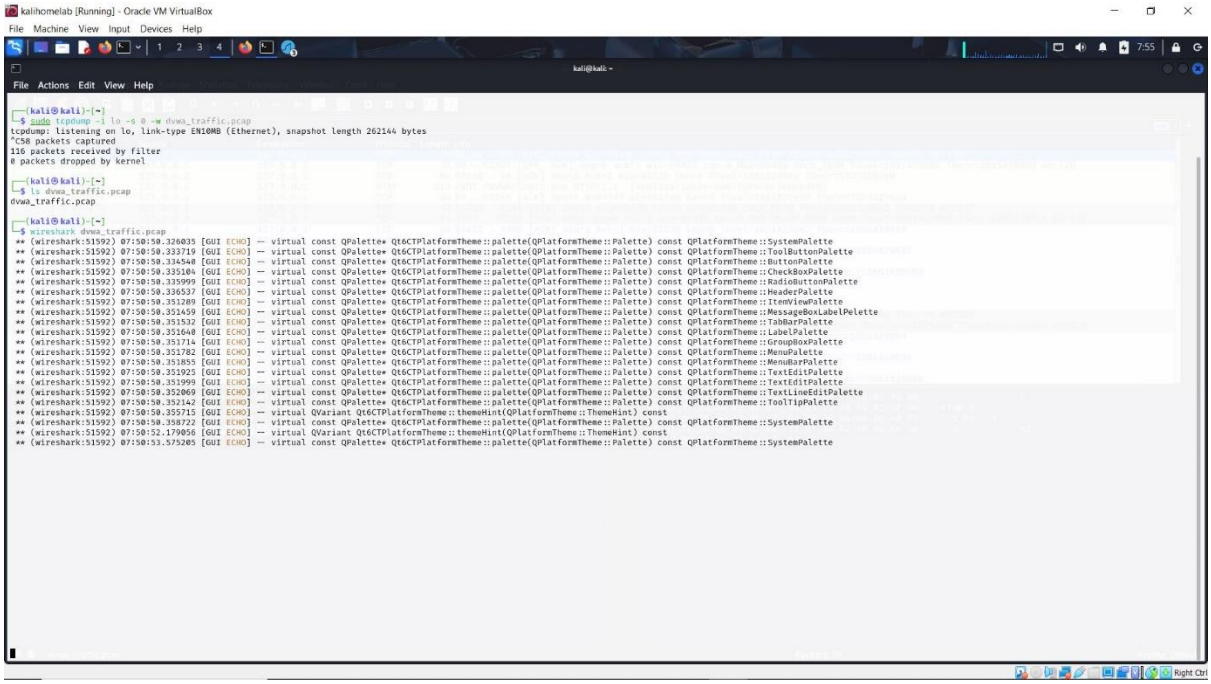
3. **Verified DVWA was running** on 127.0.0.1 (localhost).



Part 2: Capturing Local HTTP Traffic with tcpdump

1. **Initiated packet capture** on the loopback interface (lo) and saved it to a .pcap file:

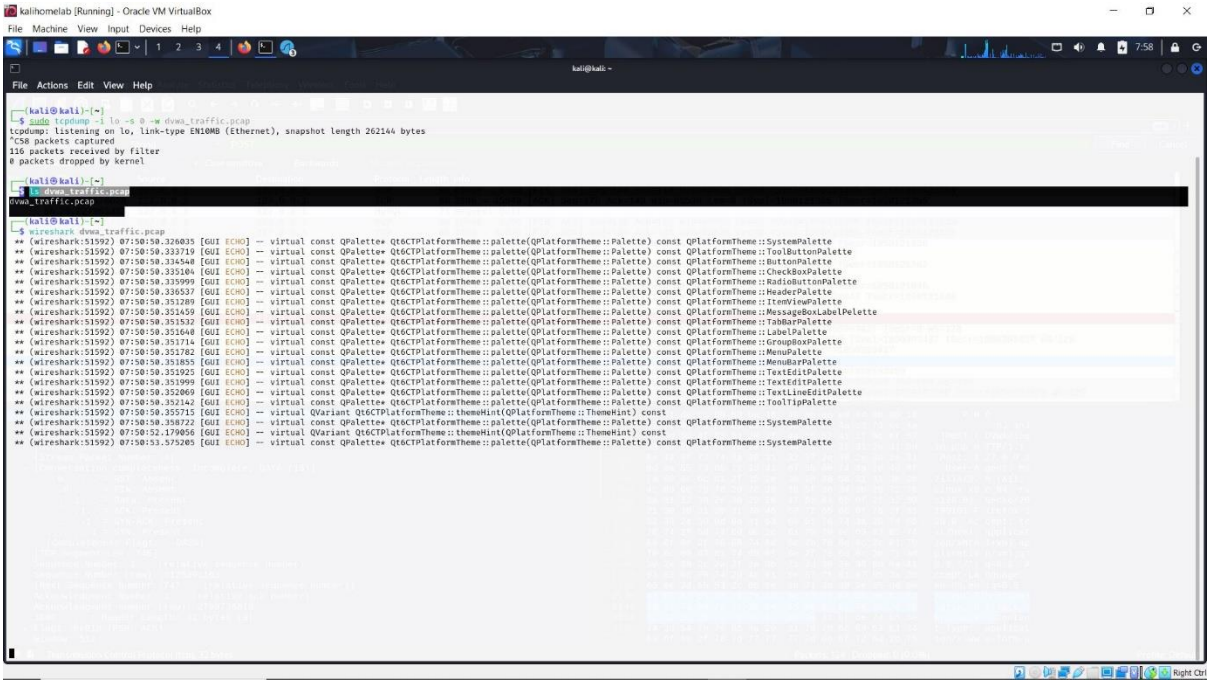
   sudo tcpdump -i lo -s 0 -w dvwa_traffic.pcap



2. **Generated HTTP traffic** by:
   - Accessing DVWA at http://127.0.0.1.
   - Logging in with default credentials (admin:password).
   - Interacting with DVWA pages (e.g., navigating to "Instructions").

3. **Stopped the capture** (CTRL+C) and verified the .pcap file:

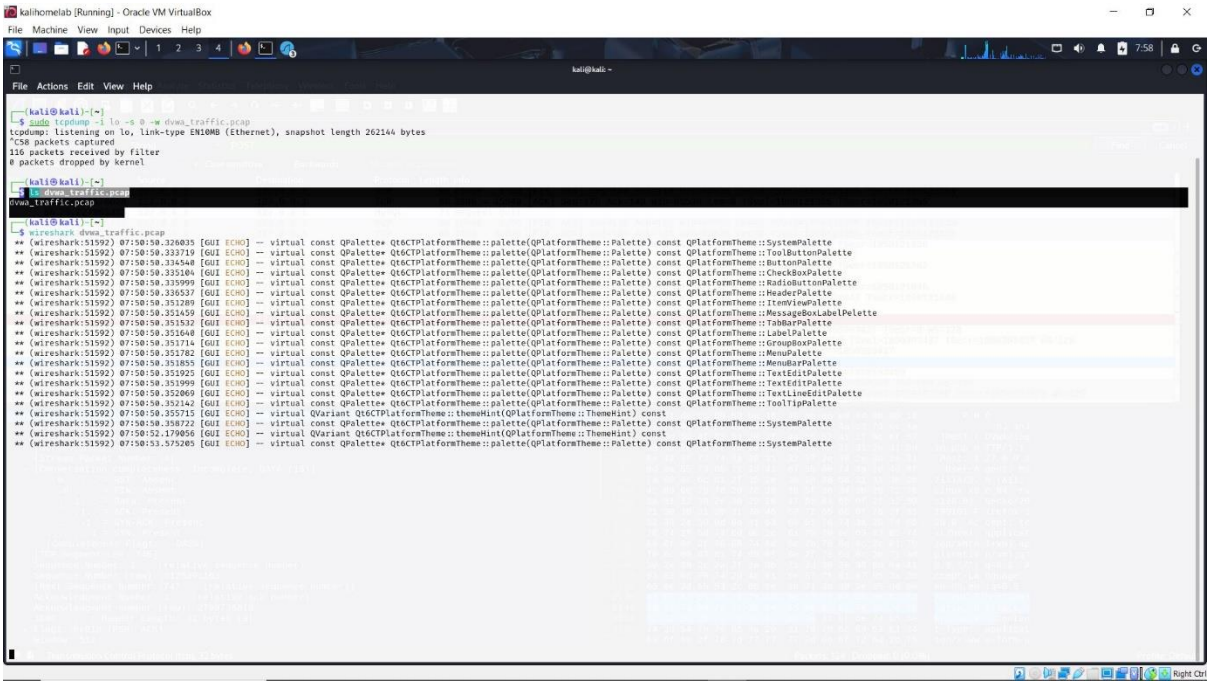
   ls dvwa_traffic.pcap



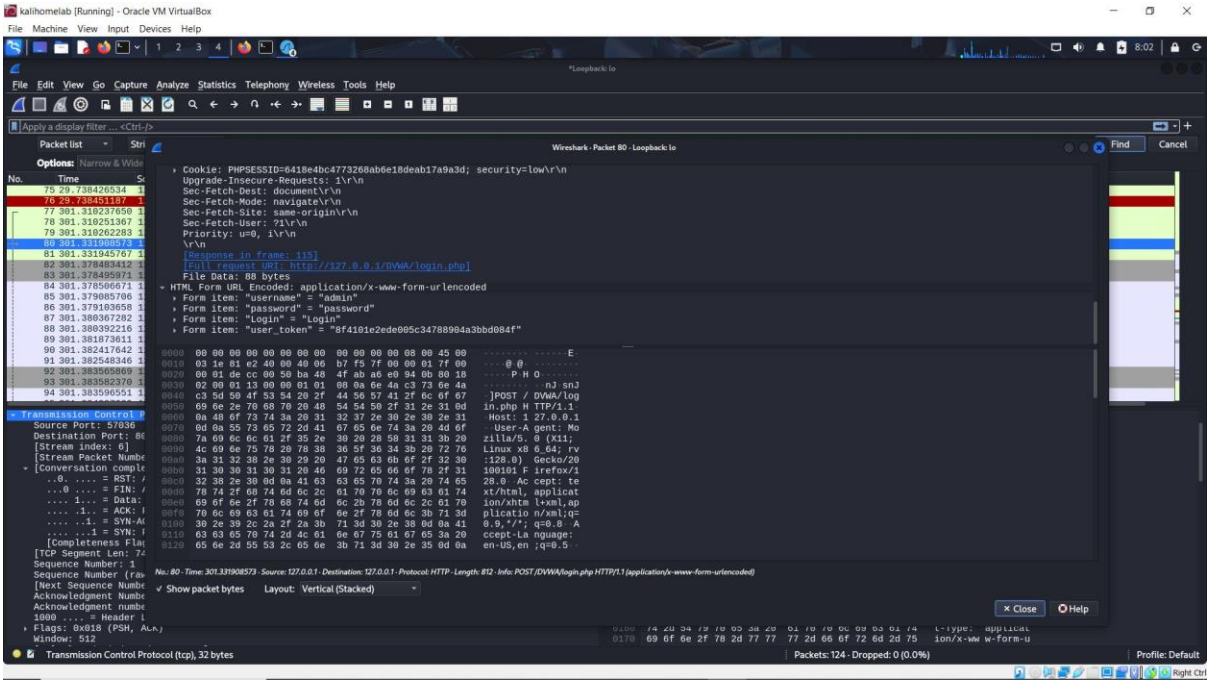
Part 3: Analysing Traffic with Wireshark

1. **Opened Wireshark** and loaded the .pcap file:

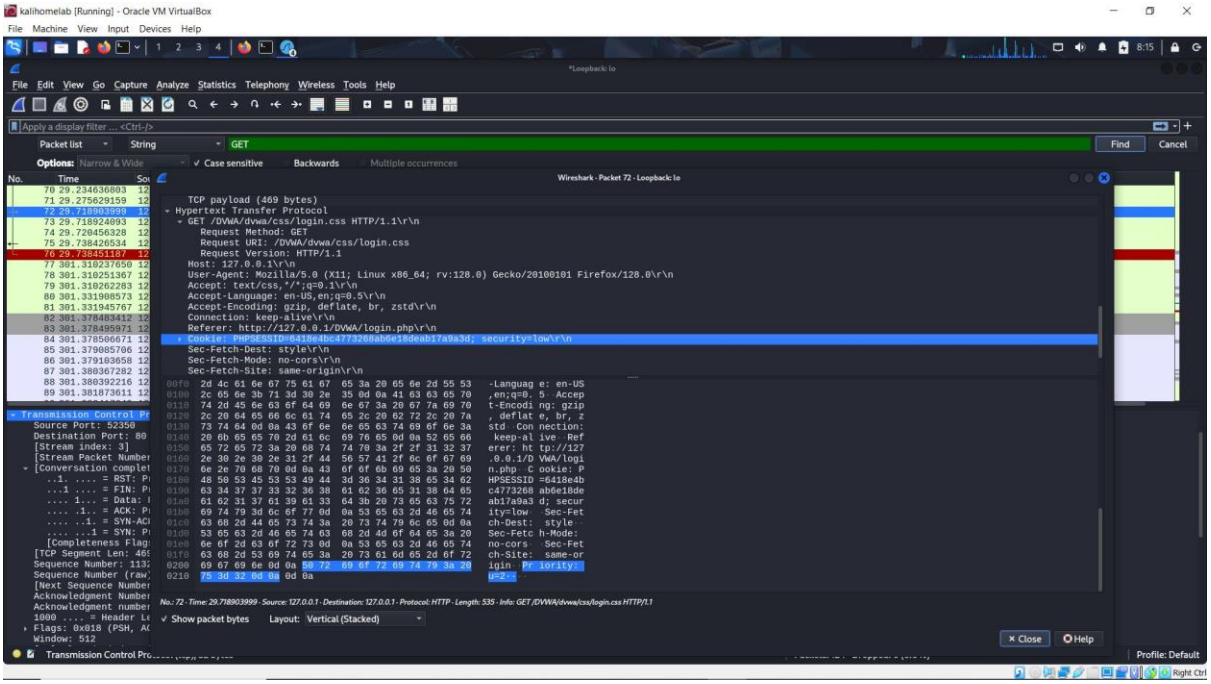
   wireshark dvwa_traffic.pcap



2. **Filtered HTTP traffic** to identify:
   - **POST requests** containing login credentials (admin:password).



   - **GET requests** revealing session cookies (PHPSESSID).



3. **Inspected HTTP headers** to:
   - Extract **plaintext credentials** from login forms.
   - Identify **session hijacking vulnerabilities** due to cookie reuse.

Key Findings

- **HTTP (unencrypted) exposed credentials** in cleartext (admin:password).
- **Session cookies (Cookie: PHPSESSID=6418e4bc477326ab6e18deab17a9a3d)** were transmitted insecurely.
- **All traffic was visible** since it ran on 127.0.0.1 (no encryption).

Reflection & Takeaways

1. **Local vs. Remote Traffic Analysis:**
    - Even on localhost, unencrypted HTTP is risky (e.g., malware could sniff traffic).
    - Tools like Wireshark can intercept **all** local communications.
2. **Security Risks Identified:**
    - **No encryption** means credentials are easily stolen.
    - **Session fixation/hijacking** is possible if cookies are intercepted.
3. **Mitigation Strategies:**
    - **Use HTTPS even for local development** (e.g., self-signed certs).
    - **Secure cookies** with HttpOnly and Secure flags.

Conclusion

This project demonstrated how **unencrypted local traffic** (even on 127.0.0.1) can expose sensitive data. It reinforced the importance of encryption and secure session management, even in development environments.

Future Work

How This Project Enhances My Portfolio

- **Real-world application** of Wireshark/tcpdump in a controlled lab.
- **Firsthand proof** of why encryption matters (even locally).
- **Vulnerability analysis** of a widely used training platform (DVWA).