HTTP-Sniffing-Wireshark

https://github.com/ahsanjunaid/HTTP-Sniffing-Wireshark

Introduction

Welcome to the Wireshark HTTP Sniffing Analysis Repository. I explored and analyzed network traffic captured with Wireshark. This repository contains packet capture files, findings, and documentation from the analysis session into HTTP communication, protocol behavior, and network troubleshooting

Setup

Virtual Machine:	Ubuntu VM hosted the Wireshark installation.
Wireshark Version:	Wireshark [version] was used for capturing and analysis.
Filtering:	Captures were filtered using the Wireshark filter http.request.method == "POST" to focus on HTTP POST requests.
Testing Site:	A testing website was used to simulate HTTP login traffic

Environment

Type:	Development
Topology:	The network was set up within a virtual environment using a virtual machine running Ubuntu.
IP Addressing:	The virtual environment utilized DHCP for IP assignment.
Security Measures:	As this was a controlled development environment, basic security measures were in place within the virtual network.
Capture Period:	Network captures were conducted during 2:30 minutes.
Purpose:	The captures were performed to analyze HTTP traffic related to a login process on a testing site.

Capture Files

Format: The primary capture file in this repository is named testing.pcap and is in the PCAP (Packet Capture) format. PCAP is a widely recognized standard for storing network packet capture data and is compatible with various network analysis tools.

File Name: Testing.pcap

Scenario: The capture file captures network traffic during various scenarios, particularly HTTP interactions related to the login process on the testing site.

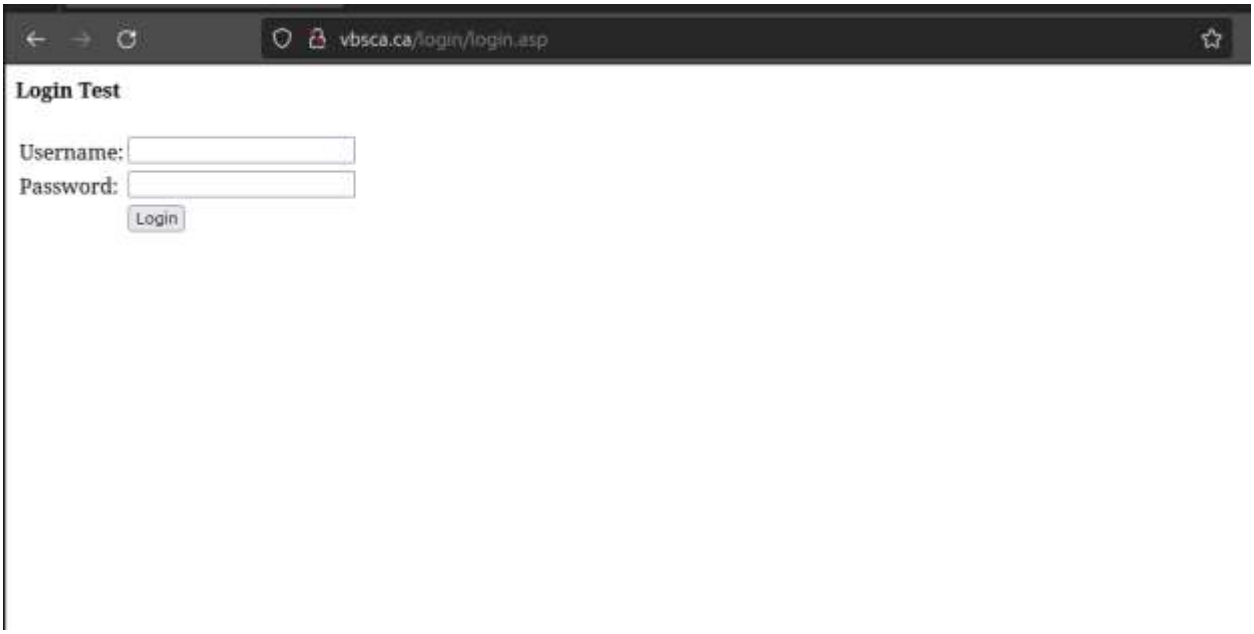
Analysis Methodology

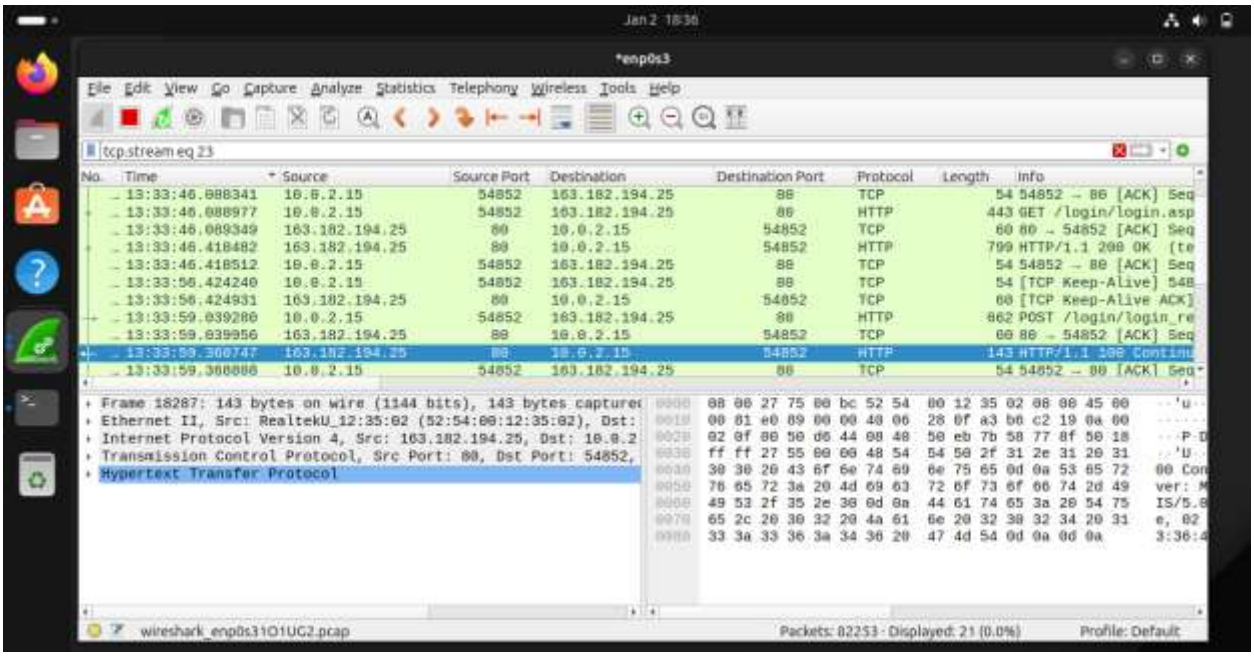
To conduct the Wireshark analysis, a controlled development environment was established using a virtual machine running Ubuntu. The primary objective was to investigate the HTTP traffic associated with a login process on a testing site. The network captures were performed using Wireshark, with a specific focus on HTTP POST requests. The Wireshark filter `http.request.method == "POST"` was applied to isolate relevant packets.

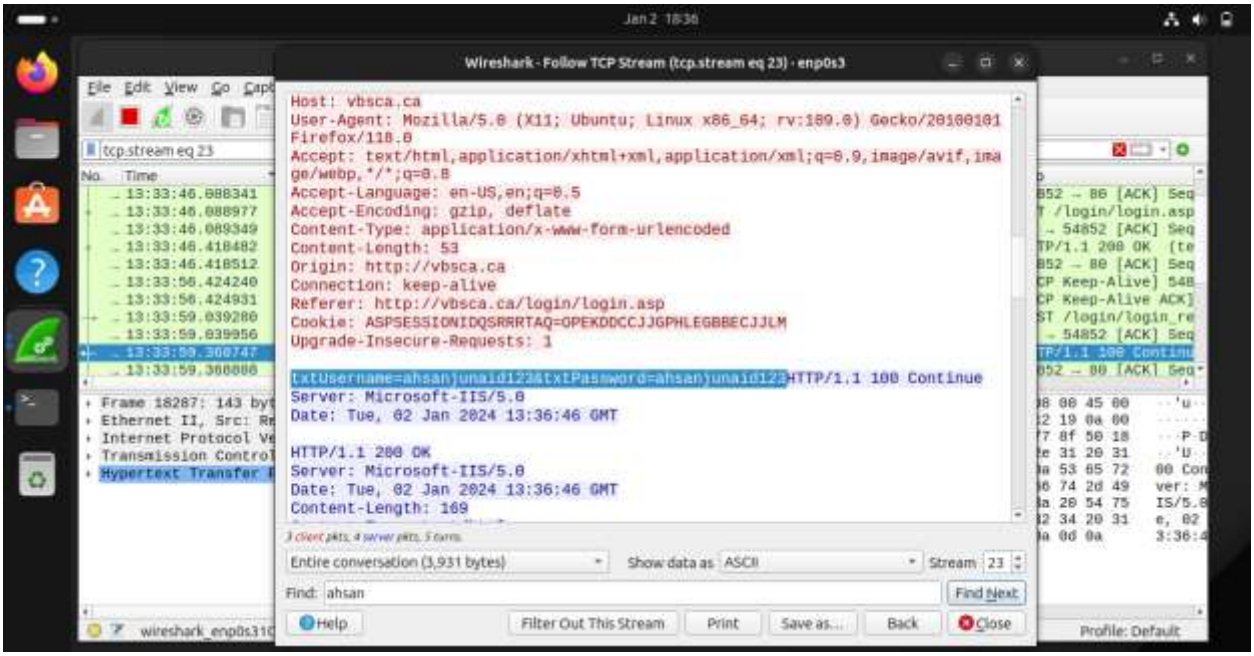
Once a suitable HTTP POST request packet was identified, further analysis was conducted by following the TCP stream associated with that packet. This process allowed for the extraction and examination of the payload exchanged during the login transaction. Through the careful inspection of the TCP stream, login credentials, including the username and password, were identified.

The testing site served as a controlled environment to simulate login activities, enabling the analysis of network traffic patterns associated with authentication processes.

Screenshots







Disclaimer and Ethical Considerations

The materials in this repository are intended for educational purposes related to network analysis. Users and contributors are expected to conduct their activities ethically, ensuring the utmost respect for privacy and adherence to legal standards. It is crucial to exercise responsible and lawful behavior when analyzing network data, refraining from the disclosure of sensitive information and maintaining a commitment to upholding ethical standards throughout the exploration and sharing of findings.