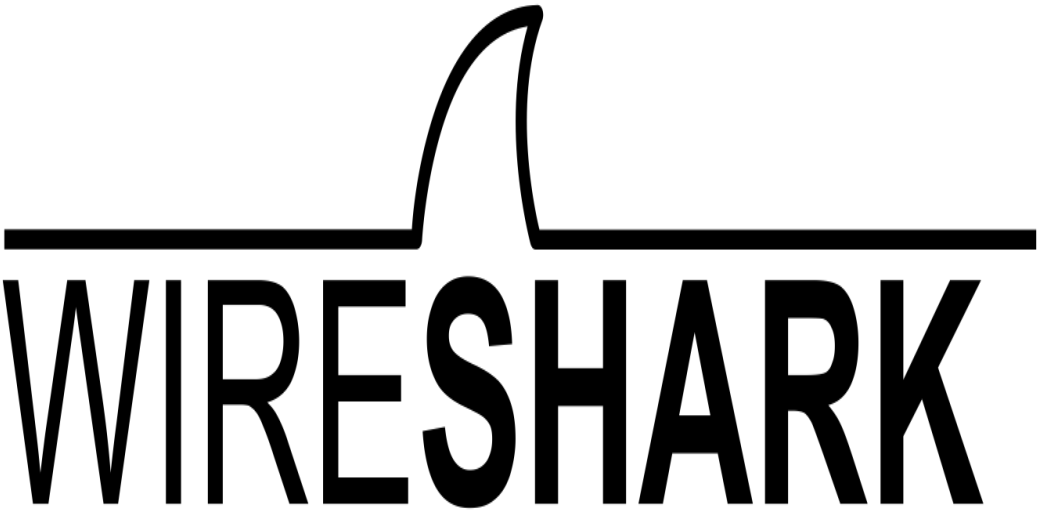


By- Charu saini

Report on wireshark

Introduction to Wireshark

Wireshark is a powerful, open-source network protocol analyzer. It enables users to capture, analyze, and troubleshoot network traffic.

Protocol Analysis

Wireshark dissects network packets, revealing the underlying protocols and data exchanged between systems.

Network Monitoring

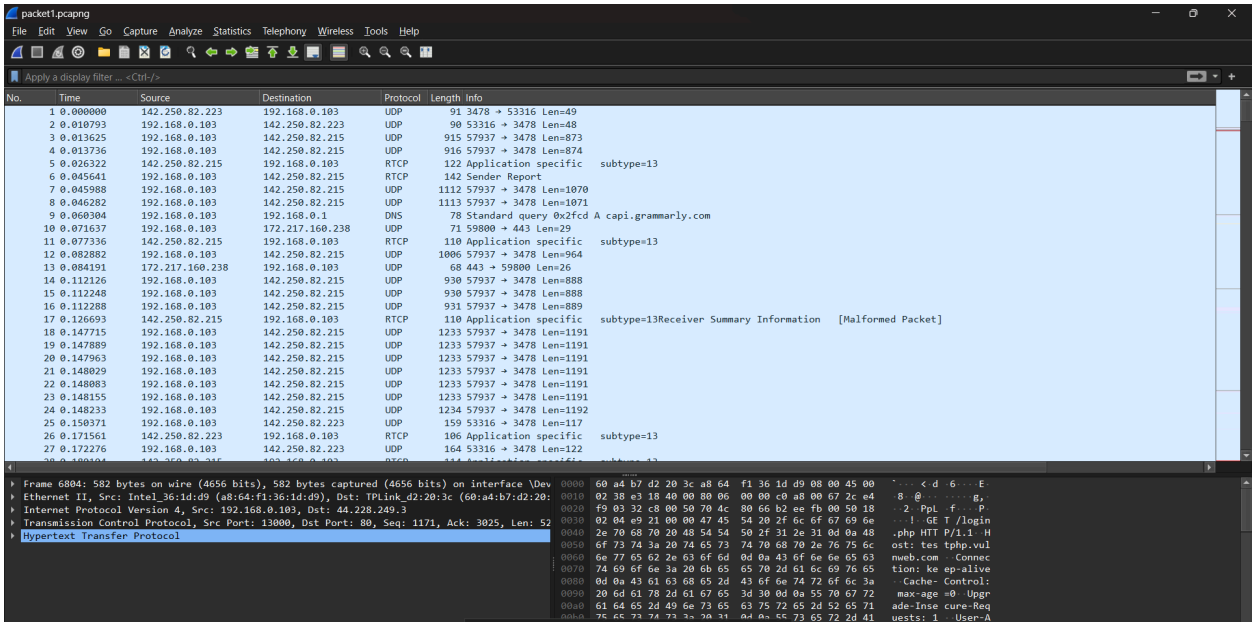
Users can monitor network traffic in real-time, identifying potential issues and security threats.

Troubleshooting

Wireshark assists in diagnosing network problems by analyzing packet flow and identifying bottlenecks or errors.

Security Auditing

Security professionals utilize Wireshark to examine network traffic for suspicious activity and identify vulnerabilities.



Wireshark UI

Key Features of Wireshark

Wireshark offers a range of features that empower users to analyze network traffic effectively.

Live Packet Capture

Wireshark allows users to capture network packets in real-time, providing an immediate view of network activity.

Packet Filtering

Users can filter captured packets based on various criteria, such as protocol, source/destination address, and port number, to focus on specific traffic.

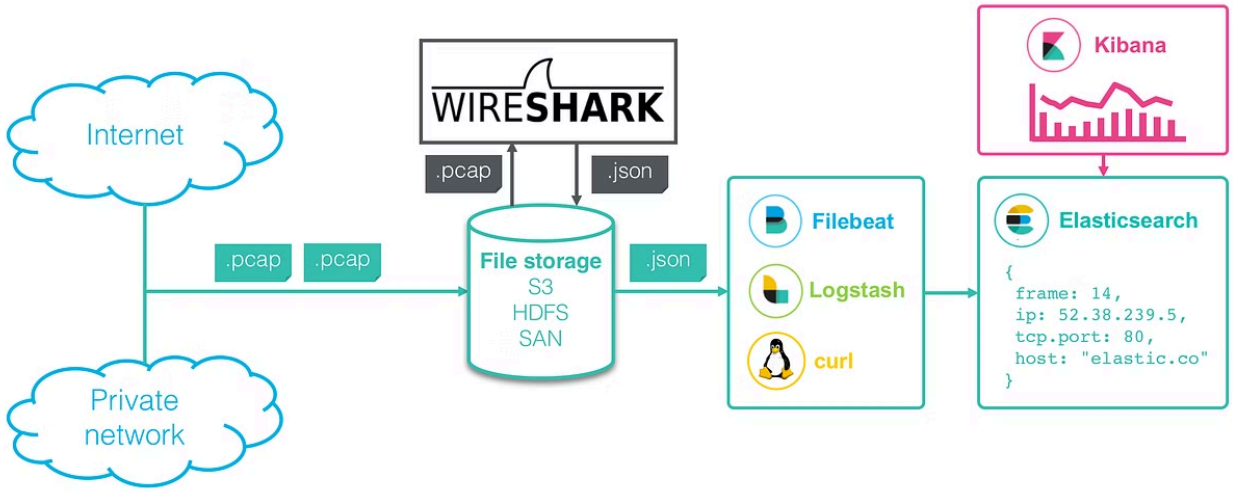
Protocol Decoding

Wireshark decodes network protocols, providing detailed information about each packet, including headers, data, and timestamps.

Traffic Analysis

Wireshark analyzes captured traffic, providing insights into network performance, security events, and application behavior.

Wireshark report



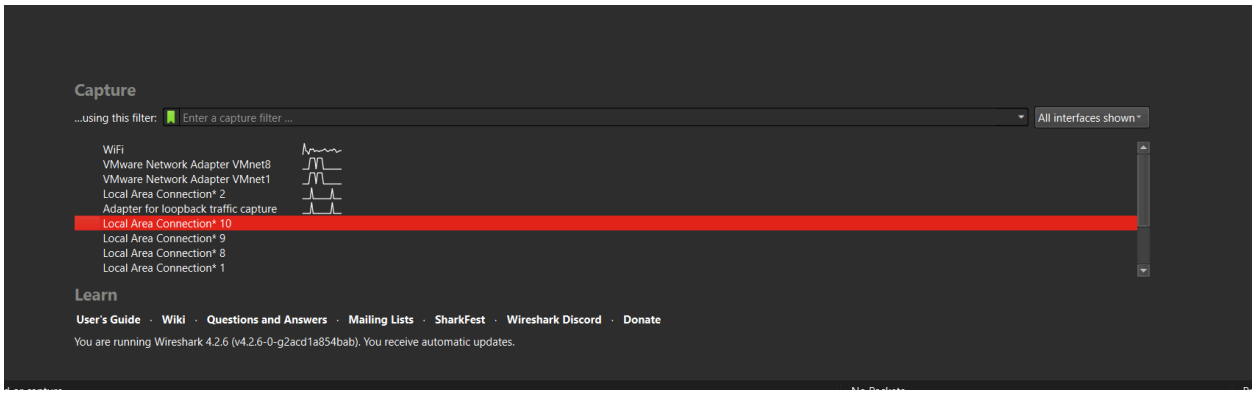
Working of wireshark

Capturing Network Traffic

Wireshark can capture network traffic from various network interfaces, including Ethernet, Wi-Fi, and virtual interfaces.

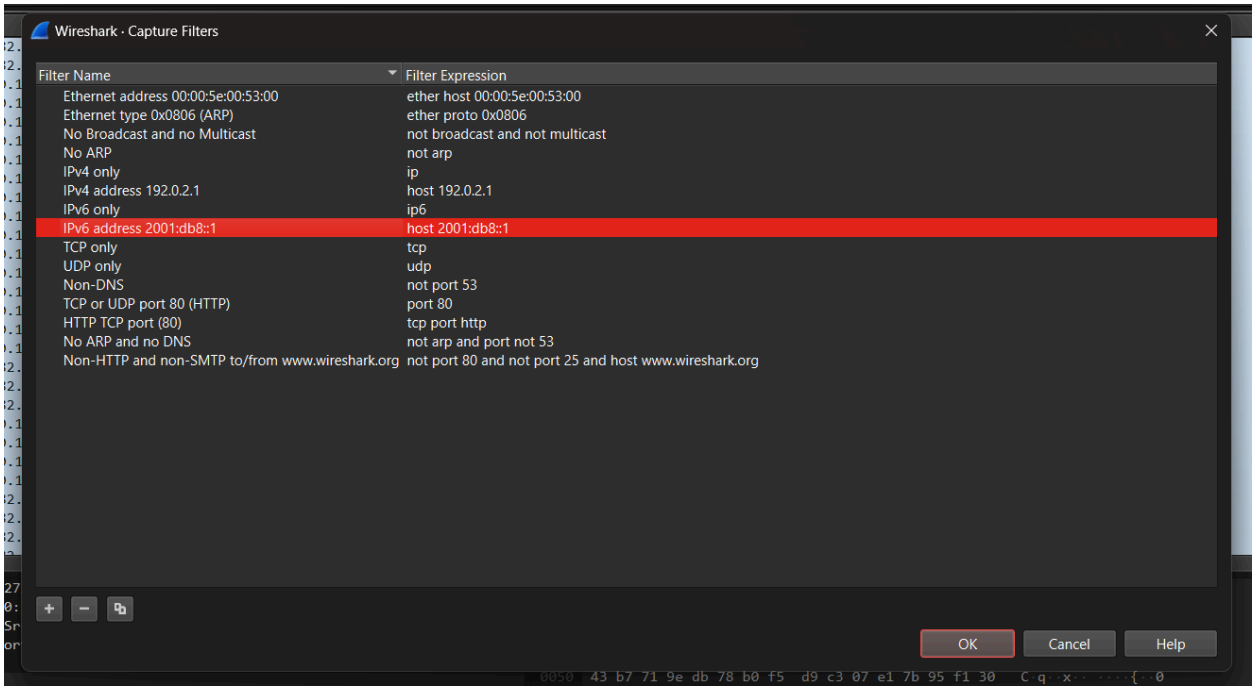
Interface Selection

Choose the network interface you want to capture traffic from.



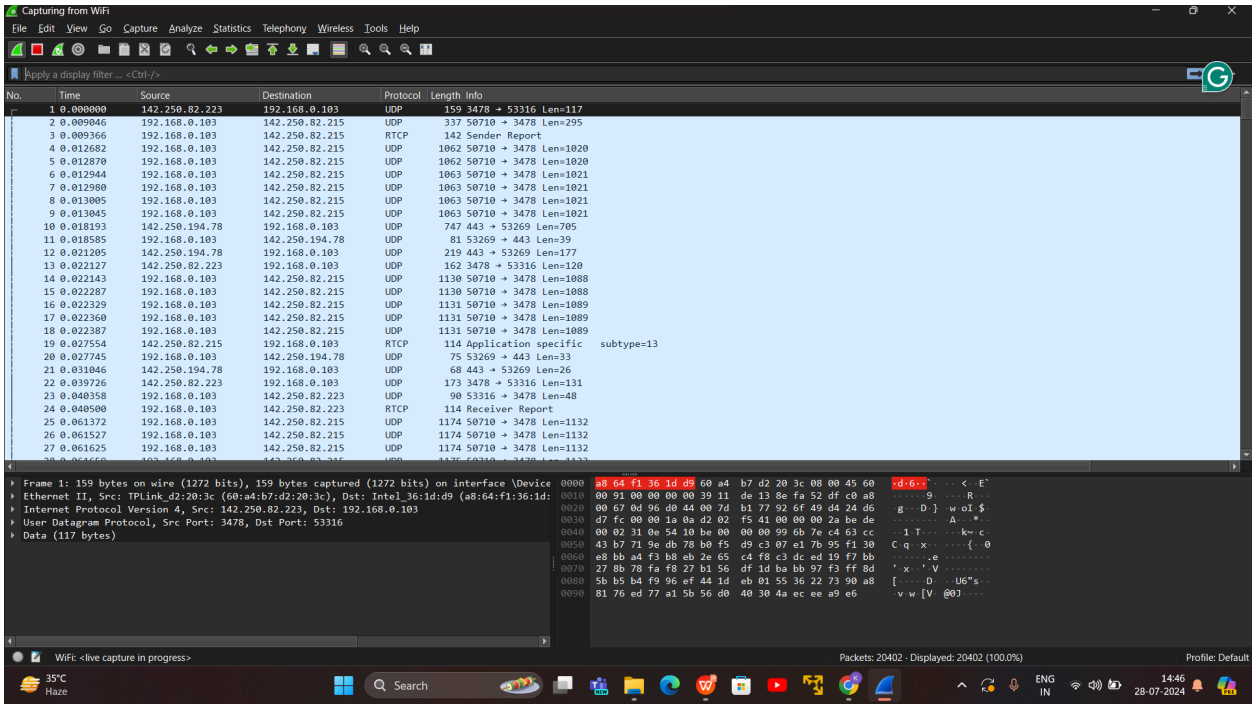
Capture Settings

Configure capture settings, such as filter, capture duration, and file format.



Start Capture

Initiate the capture process, and Wireshark will begin capturing packets.

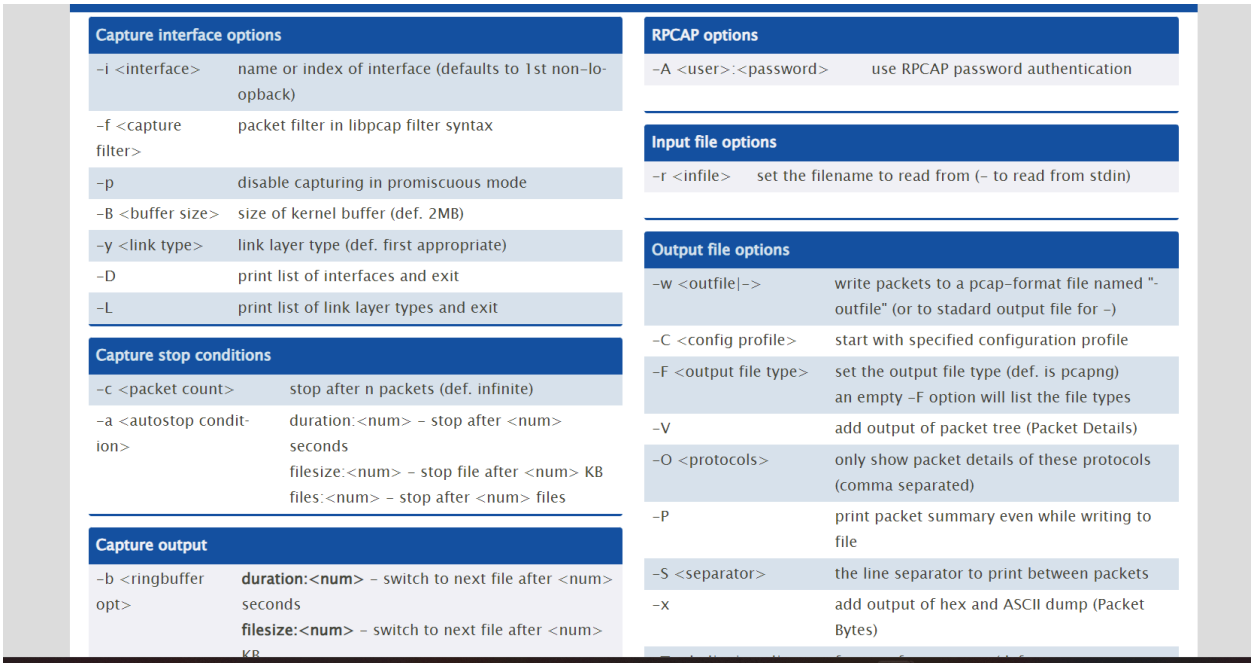


Filtering and Analyzing Packets

Wireshark offers powerful filtering capabilities to analyze specific network traffic.

Filter Syntax

Wireshark uses a flexible filter syntax based on the display filter language, allowing users to create complex filters.



Capture interface options

-i <interface>	name or index of interface (defaults to 1st non-loopback)
-f <capture filter>	packet filter in libpcap filter syntax
-p	disable capturing in promiscuous mode
-B <buffer size>	size of kernel buffer (def. 2MB)
-y <link type>	link layer type (def. first appropriate)
-D	print list of interfaces and exit
-L	print list of link layer types and exit

Capture stop conditions

-c <packet count>	stop after n packets (def. infinite)
-a <autostop condition>	duration:<num> - stop after <num> seconds filesize:<num> - stop file after <num> KB files:<num> - stop after <num> files

Capture output

-b <ringbuffer opt>	**duration:<num>** - switch to next file after <num> seconds **filesize:<num>** - switch to next file after <num> KB

RPCAP options

-A <user>:<password>	use RPCAP password authentication

Input file options

-r <infile>	set the filename to read from (- to read from stdin)

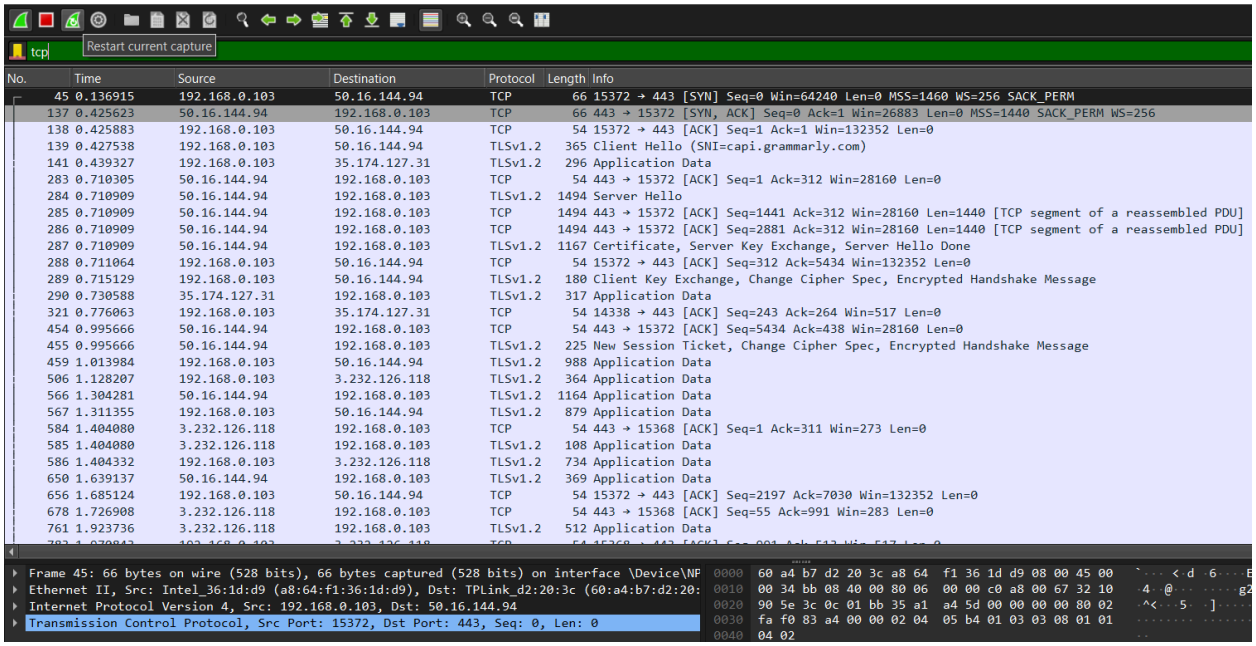
Output file options

-w <outfile|->	write packets to a pcap-format file named "-outfile" (or to stadard output file for -)
-C <config profile>	start with specified configuration profile
-F <output file type>	set the output file type (def. is pcapng) an empty -F option will list the file types
-V	add output of packet tree (Packet Details)
-O <protocols>	only show packet details of these protocols (comma separated)
-P	print packet summary even while writing to file
-S <separator>	the line separator to print between packets
-x	add output of hex and ASCII dump (Packet Bytes)

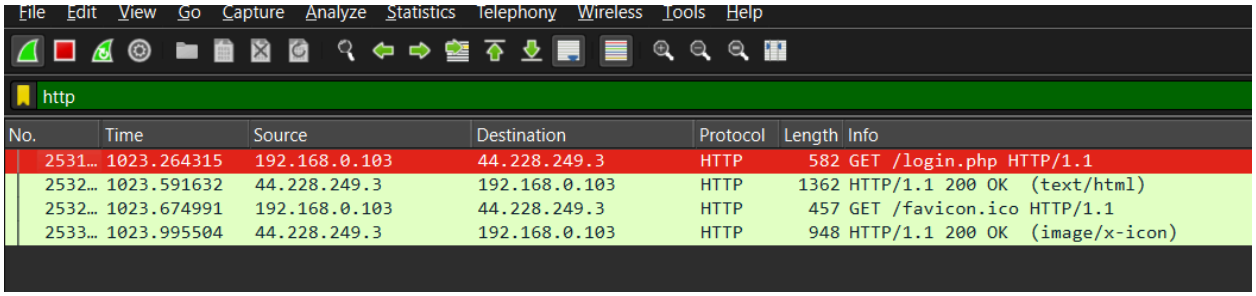
Filter Syntax

Filter Examples

Filter for specific protocols (e.g., "tcp"), source/destination addresses (e.g., "ip.addr == 192.168.1.100"), ports (e.g., "tcp.port == 80"), and more.



TCP filter



HTTP filter

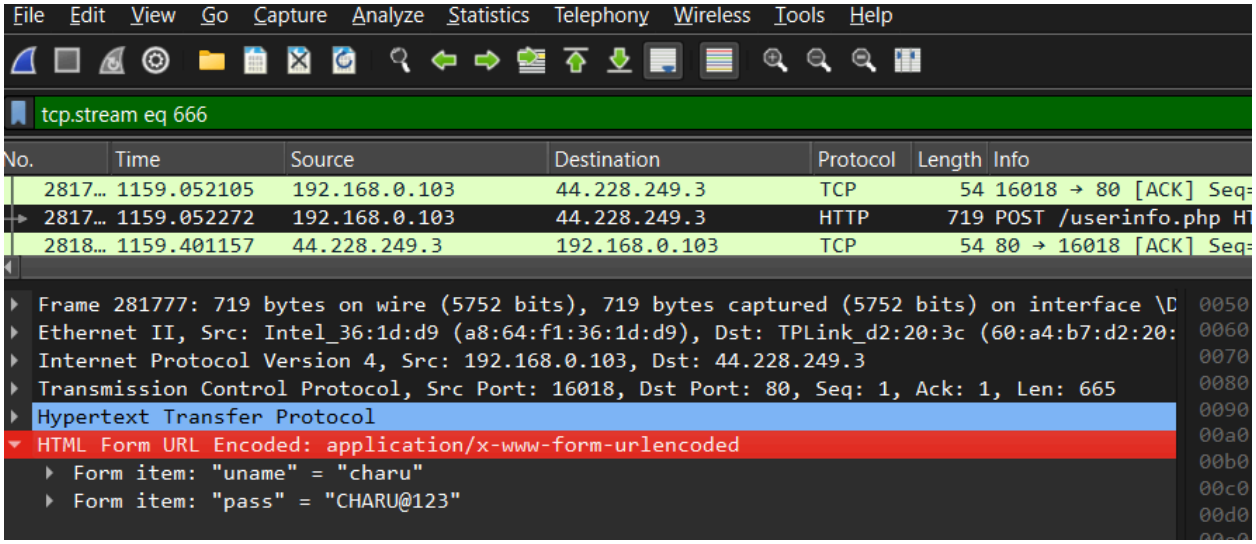
HTTP Packet Analysis

Once filtered, users can examine captured packets in detail, exploring their contents, headers, and timestamps.

Url:- http://testphp.vulnweb.com/login.php

Wireshark filter:- http.request.method == POST

Clear text credentials were captured.



Http post request in wireshark

Decoding Network Protocols

Wireshark supports decoding a wide range of network protocols, providing insights into how applications and devices communicate.

Protocol Dissecting

Wireshark breaks down network packets into their constituent parts, revealing the protocol structure and data.

Protocol Hierarchy

Wireshark displays the protocol stack, showing the layers of protocols involved in communication.

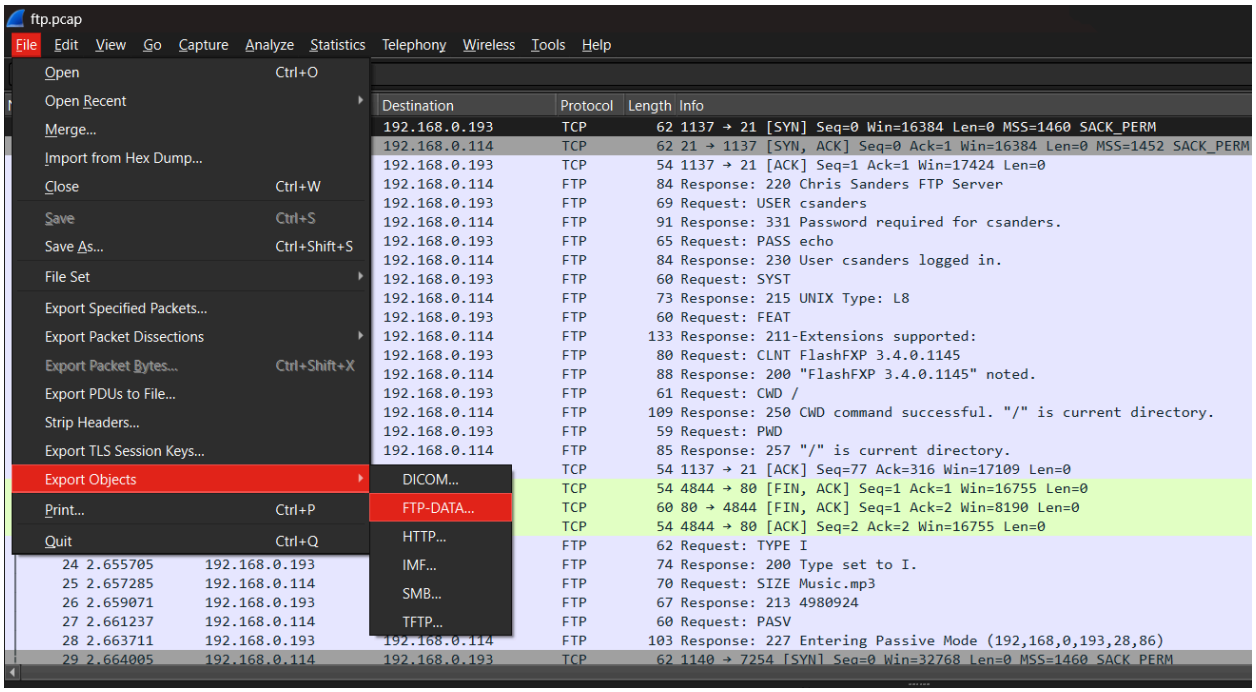
Protocol Detail

Users can explore the details of each protocol, including header fields, data payloads, and error codes.

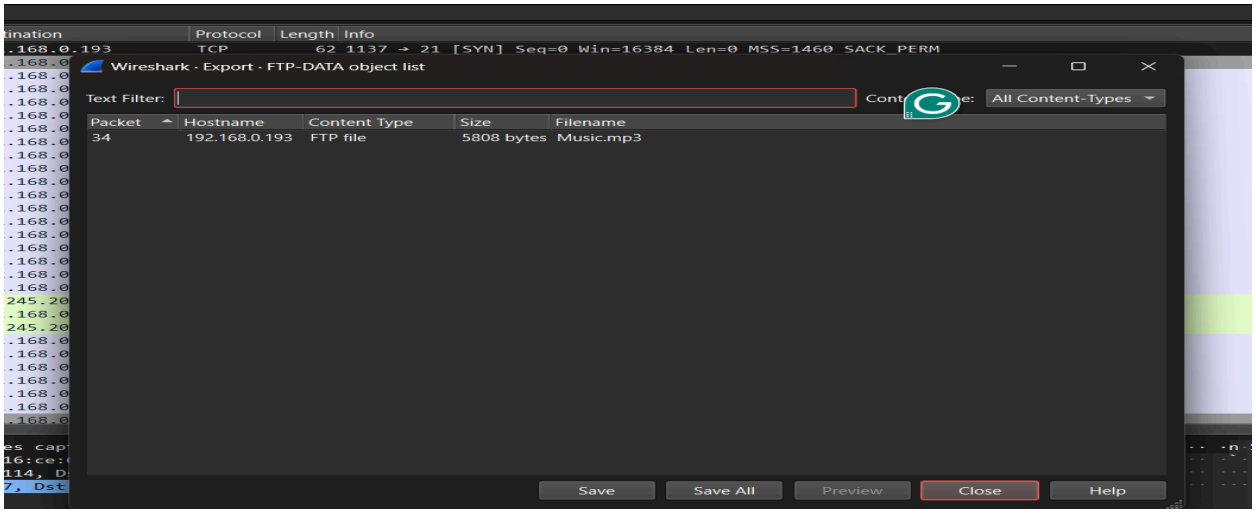
Extracting files

FTP
Extracting FTP data from wireshark

To extract files in Wireshark, select File > Export Objects. You can then select the object type you want to export based on the protocol used to transmit said object. In this case, an music file was transferred using FTP.



Exporting object



Extracting files in wireshark

Wireshark Security Capabilities

Wireshark assists security professionals in identifying and investigating security threats.

Wireshark report

Malware Detection

Analyze network traffic for malicious activity, such as botnet communication, data exfiltration, or command-and-control signals.

Security Auditing

Review network traffic to identify security vulnerabilities, misconfigurations, and unauthorized access attempts.

Intrusion Detection

Monitor network traffic for suspicious patterns, such as brute force attacks, SQL injection attempts, or denial-of-service attacks.

Conclusion

Wireshark is a powerful open-source network protocol analyzer that turns bytes on the wire into network traffic you can analyze. Its simple-to-use interface provides an overview of your capture traffic in the list pane and specific information about each packet in the details pane.