

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

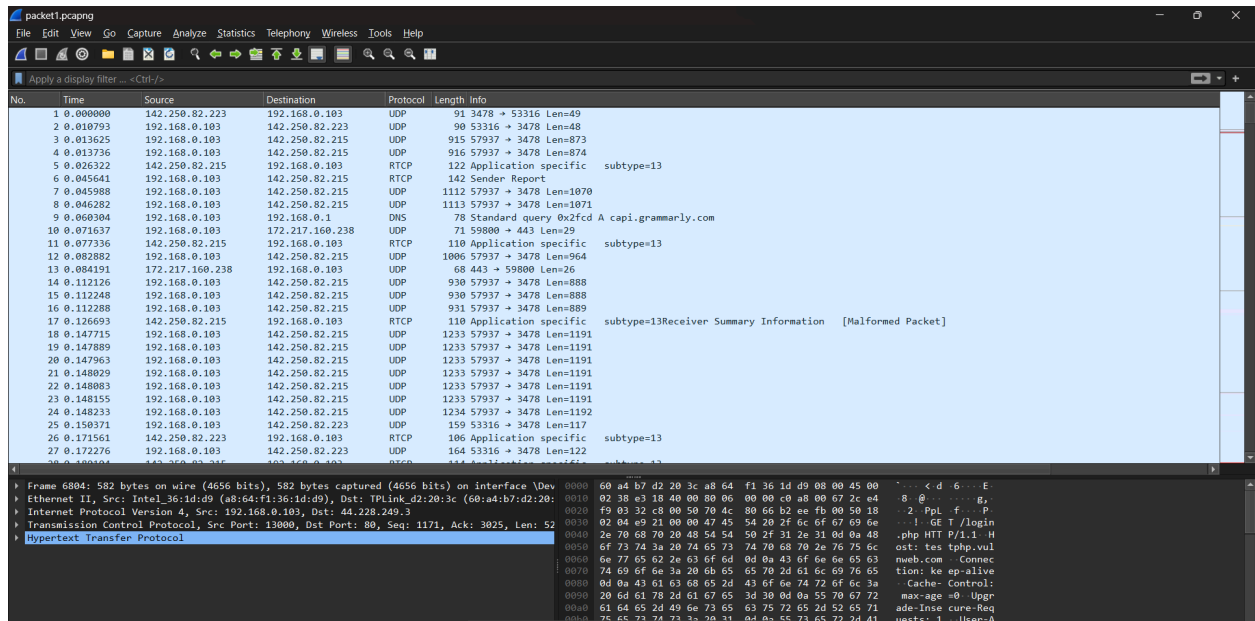
Users can monitor network traffic in real-time, identifying potential issues and security threats.

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

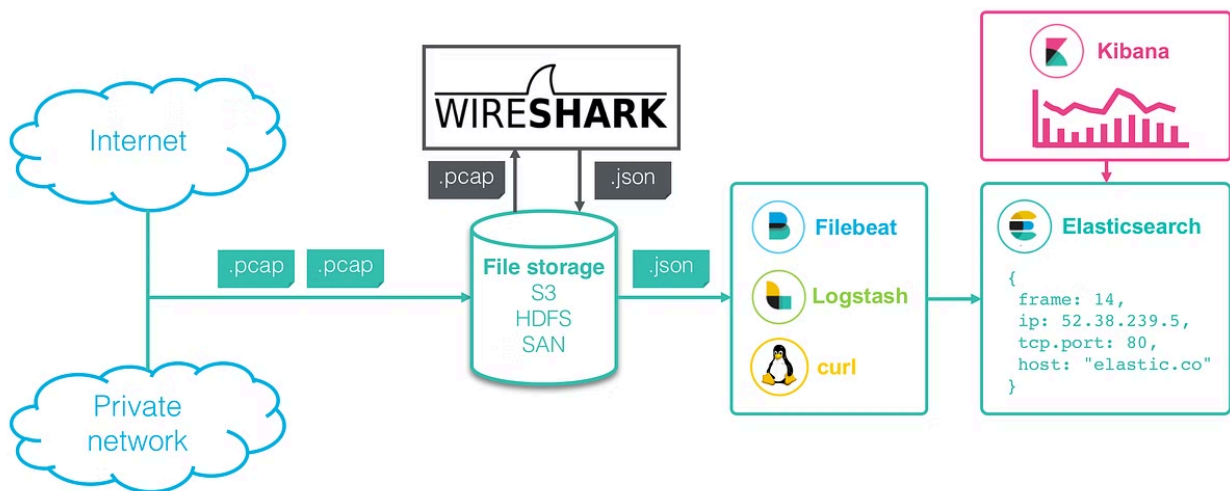
Users can filter captured packets based on various criteria, such as protocol, source/destination address, and port number, to focus on specific traffic.

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.



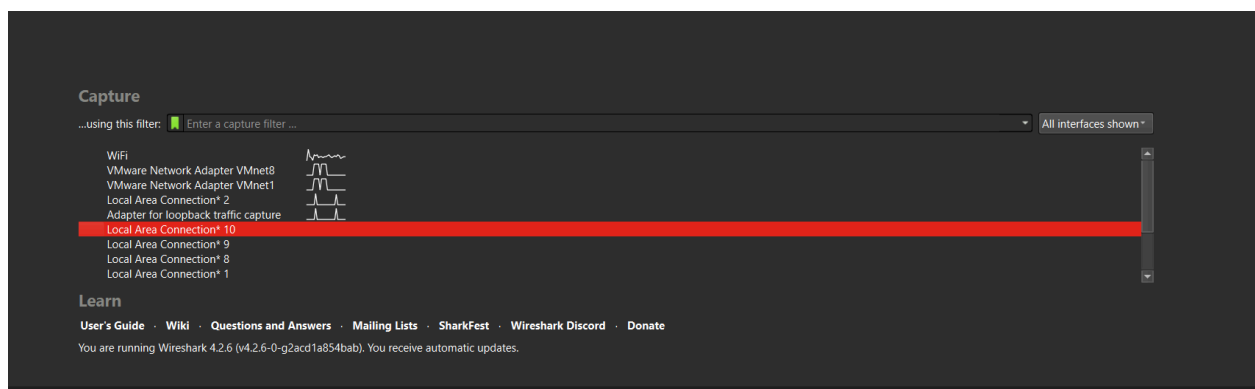
Working of wireshark

Capturing Network Traffic

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

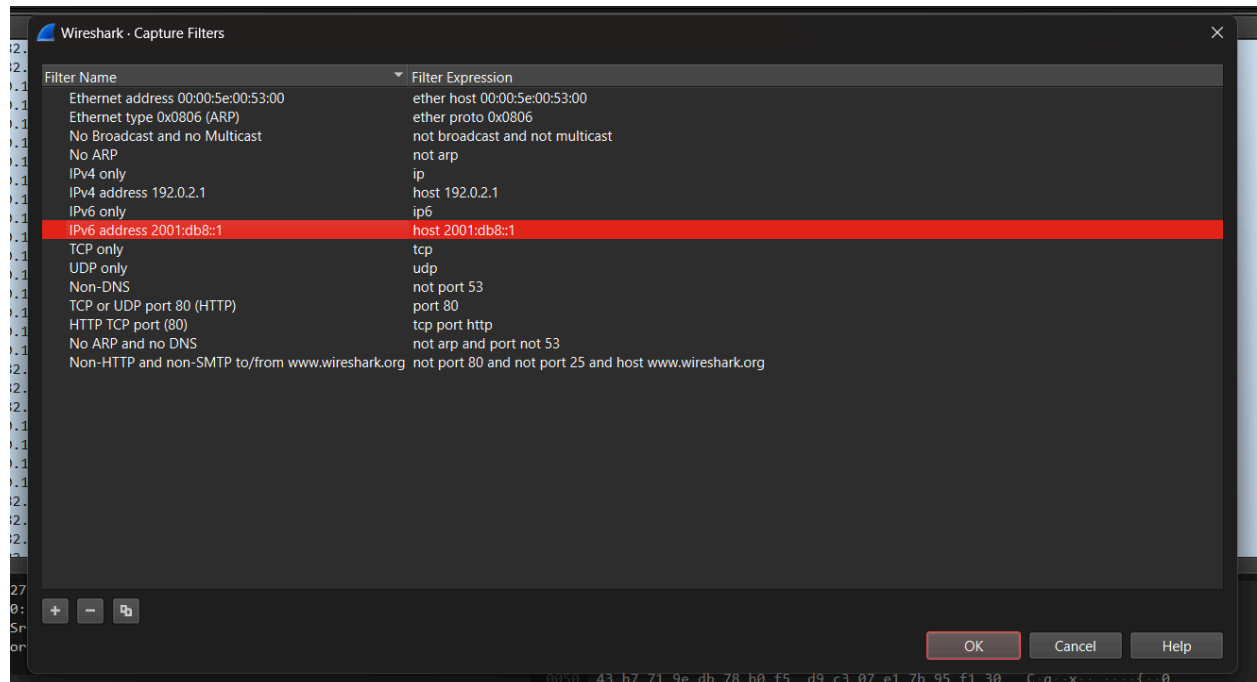
Interface Selection

Choose the network interface you want to capture traffic from.



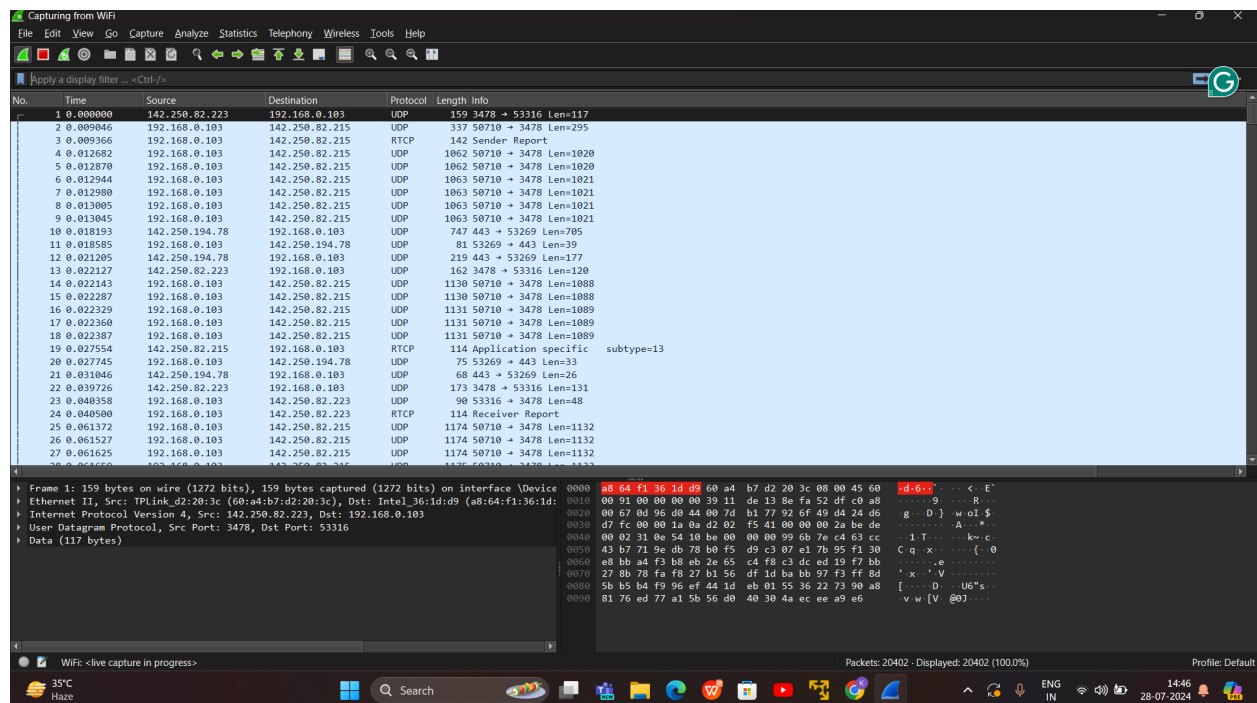
Capture Settings

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



Start Capture

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



Wireshark report

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	RPCAP options
-i <interface> name or index of interface (defaults to 1st non-loopback)	-A <user>:<password> use RPCAP password authentication
-f <capture filter> packet filter in libpcap filter syntax	
-p disable capturing in promiscuous mode	Input file options
-B <buffer size> size of kernel buffer (def. 2MB)	-r <infile> set the filename to read from (- to read from stdin)
-y <link type> link layer type (def. first appropriate)	
-D print list of interfaces and exit	Output file options
-L print list of link layer types and exit	-w <outfile -> write packets to a pcap-format file named "-outfile" (or to standard output file for -)
Capture stop conditions	-C <config profile> start with specified configuration profile
-c <packet count> stop after n packets (def. infinite)	-F <output file type> set the output file type (def. is pcapng) an empty -F option will list the file types
-a <autostop condition> duration:<num> - stop after <num> seconds filesize:<num> - stop file after <num> KB files:<num> - stop after <num> files	-V add output of packet tree (Packet Details)
Capture output	-O <protocols> only show packet details of these protocols (comma separated)
-b <ringbuffer opt> duration:<num> - switch to next file after <num> seconds filesize:<num> - switch to next file after <num> KB	-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.

No.	Time	Source	Destination	Protocol	Length	Info
45	0.136915	192.168.0.103	50.16.144.94	TCP	66	15372 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
137	0.425623	50.16.144.94	192.168.0.103	TCP	66	443 → 15372 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1440 SACK_PERM WS=256
138	0.425883	192.168.0.103	50.16.144.94	TCP	54	15372 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
139	0.427538	192.168.0.103	50.16.144.94	TLSv1.2	365	Client Hello (SNI=capi.grammarly.com)
141	0.439327	192.168.0.103	35.174.127.31	TLSv1.2	296	Application Data
283	0.710305	50.16.144.94	192.168.0.103	TCP	54	443 → 15372 [ACK] Seq=1 Ack=312 Win=28160 Len=0
284	0.710909	50.16.144.94	192.168.0.103	TLSv1.2	1494	Server Hello
285	0.710909	50.16.144.94	192.168.0.103	TCP	1494	443 → 15372 [ACK] Seq=1441 Ack=312 Win=28160 Len=1440 [TCP segment of a reassembled PDU]
286	0.710909	50.16.144.94	192.168.0.103	TCP	1494	443 → 15372 [ACK] Seq=2881 Ack=312 Win=28160 Len=1440 [TCP segment of a reassembled PDU]
287	0.710909	50.16.144.94	192.168.0.103	TLSv1.2	1167	Certificate, Server Key Exchange, Server Hello Done
288	0.711064	192.168.0.103	50.16.144.94	TCP	54	15372 → 443 [ACK] Seq=312 Ack=5434 Win=132352 Len=0
289	0.715129	192.168.0.103	50.16.144.94	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
290	0.730588	35.174.127.31	192.168.0.103	TLSv1.2	317	Application Data
321	0.776063	192.168.0.103	35.174.127.31	TCP	54	14338 → 443 [ACK] Seq=243 Ack=264 Win=517 Len=0
454	0.995666	50.16.144.94	192.168.0.103	TCP	54	443 → 15372 [ACK] Seq=5434 Ack=438 Win=28160 Len=0
455	0.995666	50.16.144.94	192.168.0.103	TLSv1.2	225	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
459	1.013984	192.168.0.103	50.16.144.94	TLSv1.2	988	Application Data
506	1.128207	192.168.0.103	3.232.126.118	TLSv1.2	364	Application Data
566	1.304281	50.16.144.94	192.168.0.103	TLSv1.2	1164	Application Data
567	1.311355	192.168.0.103	50.16.144.94	TLSv1.2	879	Application Data
584	1.404080	3.232.126.118	192.168.0.103	TCP	54	443 → 15368 [ACK] Seq=1 Ack=311 Win=273 Len=0
585	1.404080	3.232.126.118	192.168.0.103	TLSv1.2	108	Application Data
586	1.404332	192.168.0.103	3.232.126.118	TLSv1.2	734	Application Data
650	1.639137	50.16.144.94	192.168.0.103	TLSv1.2	369	Application Data
656	1.685124	192.168.0.103	50.16.144.94	TCP	54	15372 → 443 [ACK] Seq=2197 Ack=7030 Win=132352 Len=0
678	1.726908	3.232.126.118	192.168.0.103	TCP	54	443 → 15368 [ACK] Seq=55 Ack=991 Win=283 Len=0
761	1.923736	3.232.126.118	192.168.0.103	TLSv1.2	512	Application Data

TCP filter

No.	Time	Source	Destination	Protocol	Length	Info
2531...	1023.264315	192.168.0.103	44.228.249.3	HTTP	582	GET /login.php HTTP/1.1
2532...	1023.591632	44.228.249.3	192.168.0.103	HTTP	1362	HTTP/1.1 200 OK (text/html)
2532...	1023.674991	192.168.0.103	44.228.249.3	HTTP	457	GET /favicon.ico HTTP/1.1
2533...	1023.995504	44.228.249.3	192.168.0.103	HTTP	948	HTTP/1.1 200 OK (image/x-icon)

HTTP filter

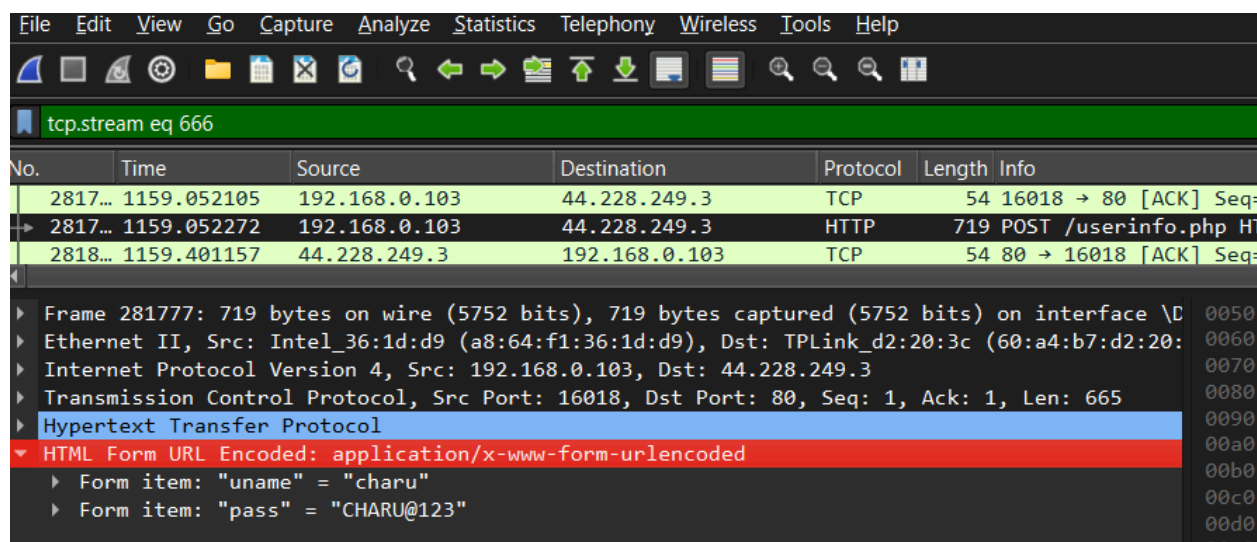
HTTP Packet Analysis

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

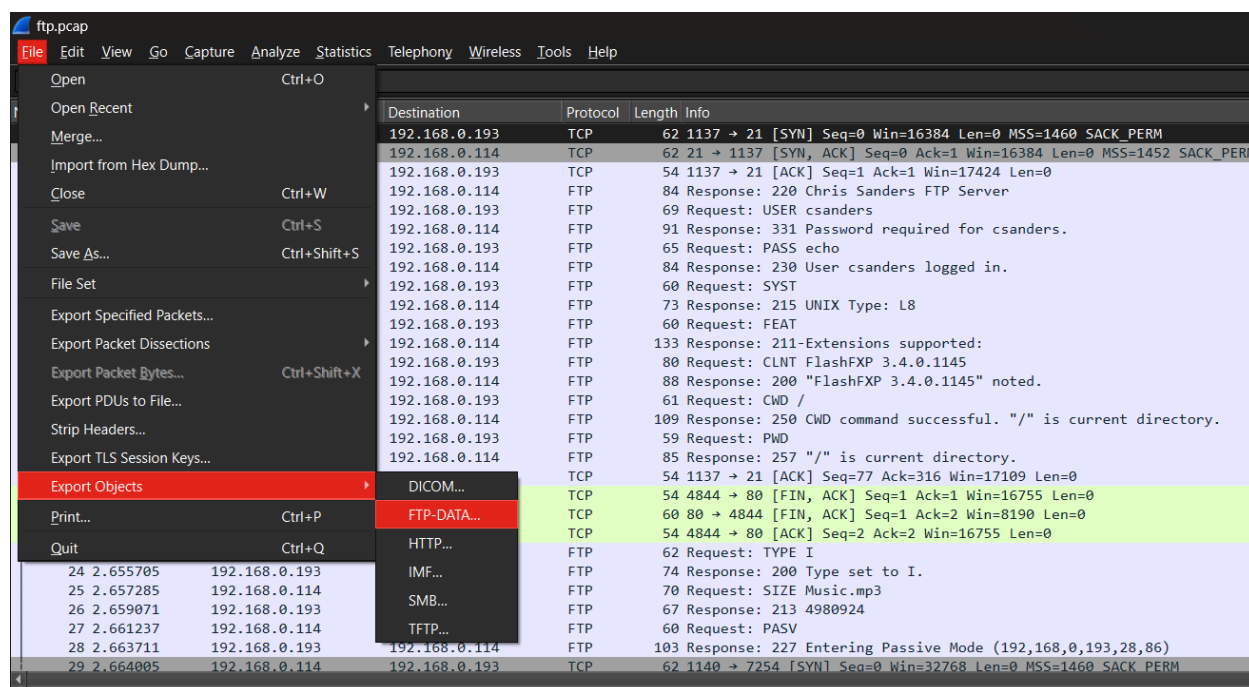
Users can explore the details of each protocol, including header fields, data payloads, and error codes.

Extracting files

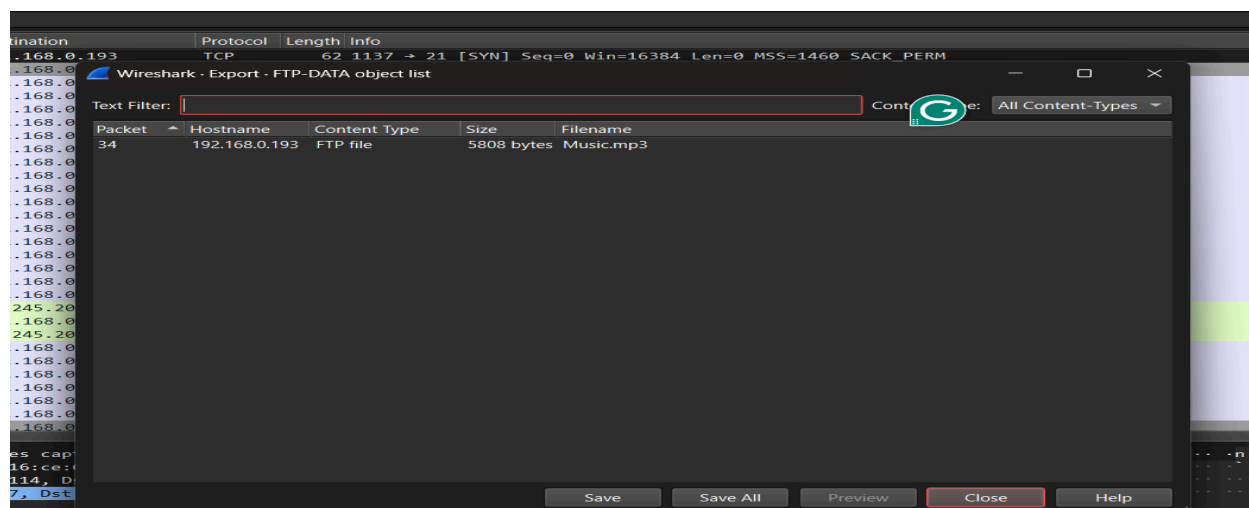
FTP

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.