

Network Traffic Forensics

Dr. Darryl J D'Souza
Mob No: 9986382162



Network Traffic Forensics



Network forensics is a subcategory of digital forensics that essentially deals with the examination of the network and its traffic going across a network that is suspected to be involved in malicious activities, and its investigation for example a network that is spreading malware for stealing credentials or for the purpose analyzing the cyber-attacks. As the internet grew cybercrimes also grew along with it and so did the significance of network forensics, with the development and acceptance of network-based services such as the World Wide Web, e-mails, and others.

In this lab, we will start with a refresher on the basics of networks, including IP addresses, protocols, and ports. We will explore how these elements work together to facilitate communication between devices.

Next, we will move on to the forensic analysis of network traffic. We will analyze captured traffic for suspicious or malicious activity, which will involve looking for anomalies in traffic patterns, identifying unauthorized access attempts, and tracing the source of any attacks.

Basic Networking Refresher

IP Addresses

An IP address is a unique identifier assigned to each device on a network. It is used to identify the device's location and facilitate communication with other devices. An example of an IP address is “192.168.1.1”.

Protocols

A protocol is a set of rules that govern the communication between devices on a network. Some common protocols include HTTP, SSH, and FTP.

For example, when you browse a website, your computer sends an HTTP request to the web server, which responds with an HTTP response that contains the webpage you requested.

Ports

A port is a number used to identify a specific application or service on a device. When a device receives network traffic, it uses the port number to determine which application or service the traffic is intended for.

For example, when you access a website using HTTP, your browser sends the request to port 80 on the web server.

Protocols and their port numbers

Protocol	Port
FTP (File Transfer Protocol)	21
SSH (Secure Shell)	22
Telnet (Teletype Network)	23
SMTP (Simple Mail Transfer Protocol)	25
DNS (Domain Name System)	53
DHCP (Dynamic Host Configuration Protocol)	67/68
HTTP (Hyper Text Transfer Protocol)	80
HTTPS (Hyper Text Transfer Protocol Secure)	443
SMB (Server Message Block)	445
RDP (Remote Desktop Protocol)	3389
ICMP (Internet Control Message Protocol)	N/A

Network Traffic Forensics

Capturing Network Traffic

To capture live network traffic, follow the steps below:

1. Open Wireshark by entering wireshark in the terminal on Linux or through the Start menu on Windows.
2. Once it is open, select a network interface that you want to capture traffic on. In most cases, it is eth0.
3. Click the blue colored shark's fin icon on top left of the window to start capturing traffic.
4. In most cases you should already see packets begin to appear on Wireshark. If not, you can try generating some traffic by visiting a website in your browser, like <https://www.google.com/>.
5. To stop capturing traffic, click the stop button in red on top left.
6. To save the capture traffic, go to File → Save as, and select a name and location to save the capture file.

Note: By default, the file extension should be .pcapng. Another commonly used extension for capture files is .pcap.

Analyzing Network Traffic

The next step in network traffic forensics is to analyze the captured network traffic. This involves examining the network traffic to identify suspicious activity, attempts for unauthorized access, and extracting information such as:

- source and destination IP addresses and ports
- protocols
- data/payload transmitted
- date and time of the activity

While there are many features available in Wireshark for network forensics, some of the most commonly used ones include viewing protocol hierarchy, applying filters, viewing packet details and packet bytes, following TCP streams, and exporting objects.

begin analyzing sample captured network traffic, download the file - Lab.pcapng

Protocol Hierarchy

To get a general overview of the captured network traffic, we can go to Statistics → Protocol Hierarchy to see which protocols are being used in the capture, and the relative amount of packets for each protocol. This helps us narrow down our analysis and filter for suspicious traffic.

Wireshark · Protocol Hierarchy Statistics · capture.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	26	100.0	77799	51 M	0	0	0	26
Ethernet	100.0	26	0.5	364	242 k	0	0	0	26
Internet Protocol Version 4	100.0	26	0.7	520	345 k	0	0	0	26
Transmission Control Protocol	100.0	26	98.9	76915	51 M	24	76269	50 M	26
Hypertext Transfer Protocol	7.7	2	97.8	76067	50 M	1	137	91 k	2
JPEG File Interchange Format	3.8	1	97.4	75741	50 M	1	75741	50 M	1

Display filter: tcp.stream eq 5

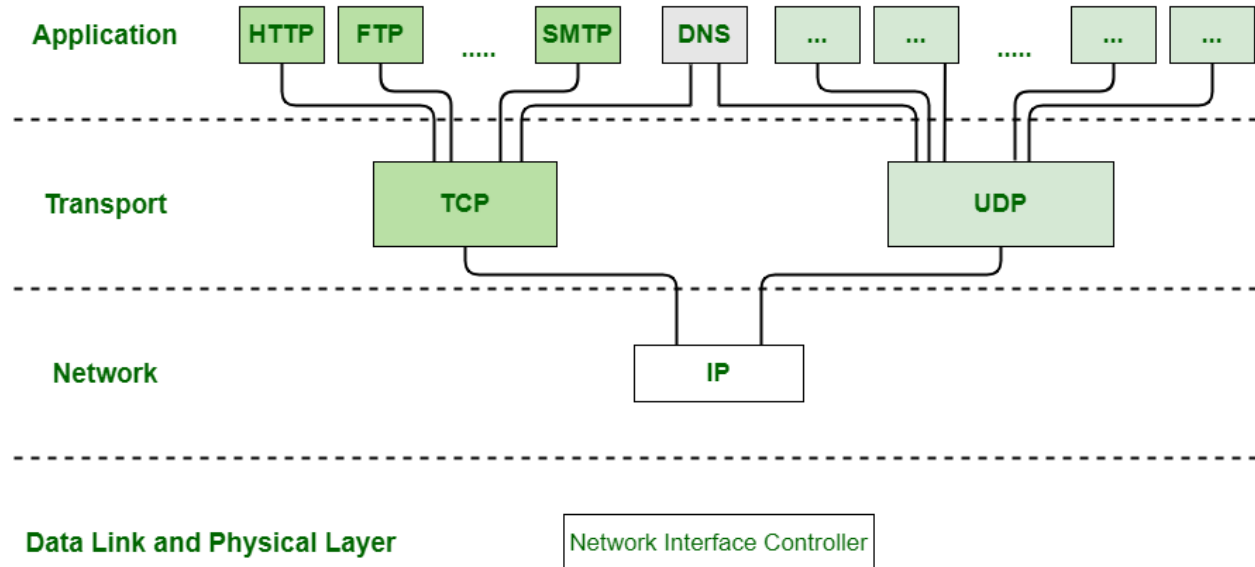
CloseCopyHelp

Protocol Hierarchies

Generally, Computer networks are comprised of or contain a large number of pieces of hardware and software. To just simplify network design, various networks are organized and arranged as a stack of layers of hardware and software, one on top of another.

The number, name, content, and function of each layer might vary and can be different from one network to another. The main purpose of each of layers is just to offer and provide services to higher layers that are present.

Each and every layer has some particular task or function. In programming, this concept is very common. The networks are organized and arranged as different layers or levels simply to reduce and minimize complexity of design of network software.



Filters



For you to Try :

Apply filters to focus on traffic of your interest. This makes it easier to analyze and view only the relevant packets that you need.

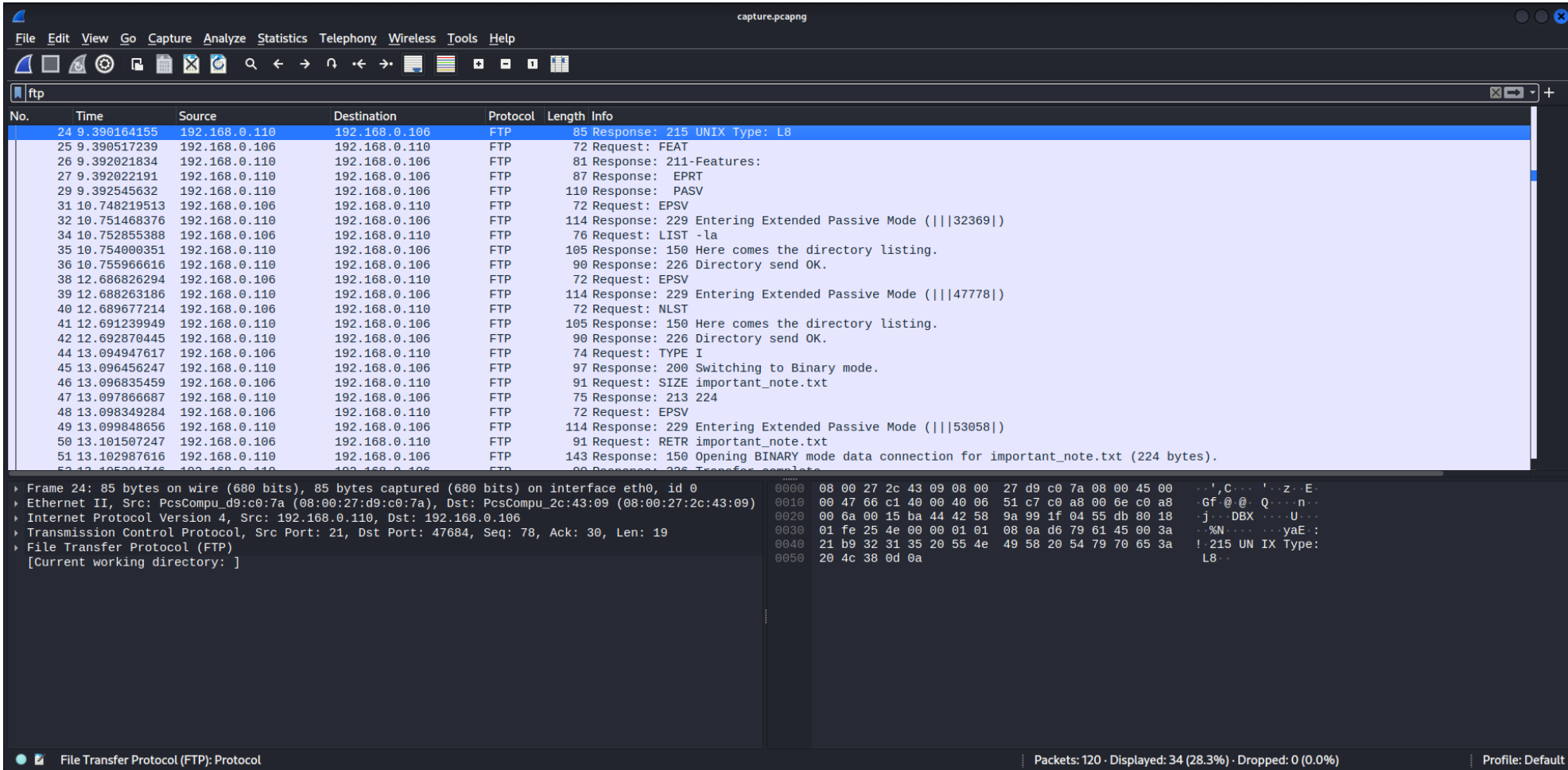
Eg: HTTP, FTP

Try other filters Wireshark supports, refer to

<https://wiki.wireshark.org/DisplayFilters>.

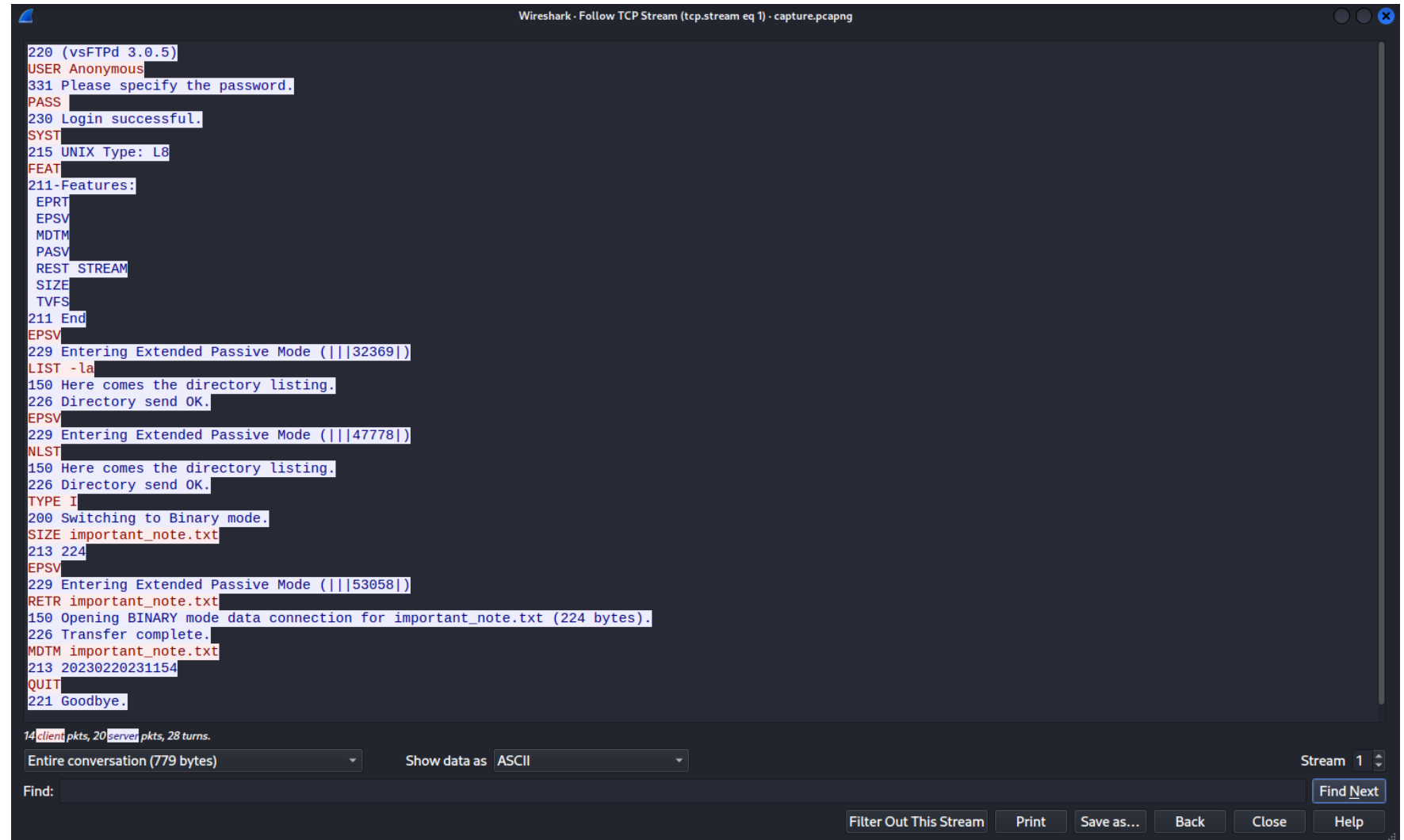
Packet Details and Packet Bytes

The packet details pane shows the selected packet in a more detailed form, whereas the packet bytes pane shows the data of the selected packet in a hexdump format. These panes are located at the bottom of the Wireshark window.



Follow TCP Stream

The Follow TCP Stream feature in Wireshark displays the entire conversation for a particular TCP connection. This makes it easier to see the full details of the connection and any data transmitted during that connection. To use this feature, we can right click on any packet, then select Follow → TCP Stream.



The image shows the 'Follow TCP Stream' window in Wireshark, titled 'Wireshark - Follow TCP Stream (tcp.stream eq 1) - capture.pcapng'. The window displays the raw data of a TCP stream, which is an FTP session. The data is color-coded: red for commands, blue for responses, and black for other data. The session starts with a 220 response from the server (vsFTPD 3.0.5). The client sends a USER command, followed by a 331 response asking for a password. The client then sends a PASS command, followed by a 230 response indicating successful login. The client sends a SYST command, followed by a 215 response indicating the UNIX Type: L8. The client sends a FEAT command, followed by a 211 response listing features: EPRT, EPSV, MDTM, PASV, REST STREAM, SIZE, and TVFS. The client then sends an EPSV command, followed by a 229 response entering extended passive mode (|||32369|). The client sends a LIST -la command, followed by a 150 response opening a data connection. The client then sends a 226 response indicating the directory listing is complete. The client sends another EPSV command, followed by a 229 response entering extended passive mode (|||47778|). The client sends another LIST -la command, followed by a 150 response opening a data connection. The client then sends a 226 response indicating the directory listing is complete. The client sends a TYPE I command, followed by a 200 response switching to binary mode. The client sends a SIZE command, followed by a 213 response indicating the file size is 224 bytes. The client then sends an EPSV command, followed by a 229 response entering extended passive mode (|||53058|). The client sends a RETR important_note.txt command, followed by a 150 response opening a binary mode data connection for important_note.txt (224 bytes). The client then sends a 226 response indicating the transfer is complete. The client sends an MDTM important_note.txt command, followed by a 213 response indicating the modification time is 20230220231154. The client then sends a QUIT command, followed by a 221 response indicating goodbye.

```
220 (vsFTPD 3.0.5)
USER Anonymous
331 Please specify the password.
PASS
230 Login successful.
SYST
215 UNIX Type: L8
FEAT
211-Features:
EPRT
EPSV
MDTM
PASV
REST STREAM
SIZE
TVFS
211 End
EPSV
229 Entering Extended Passive Mode (|||32369|)
LIST -la
150 Here comes the directory listing.
226 Directory send OK.
EPSV
229 Entering Extended Passive Mode (|||47778|)
NLST
150 Here comes the directory listing.
226 Directory send OK.
TYPE I
200 Switching to Binary mode.
SIZE important_note.txt
213 224
EPSV
229 Entering Extended Passive Mode (|||53058|)
RETR important_note.txt
150 Opening BINARY mode data connection for important_note.txt (224 bytes).
226 Transfer complete.
MDTM important_note.txt
213 20230220231154
QUIT
221 Goodbye.
```

14 client pkts, 20 server pkts, 28 turns.

Entire conversation (779 bytes) Show data as ASCII Stream 1

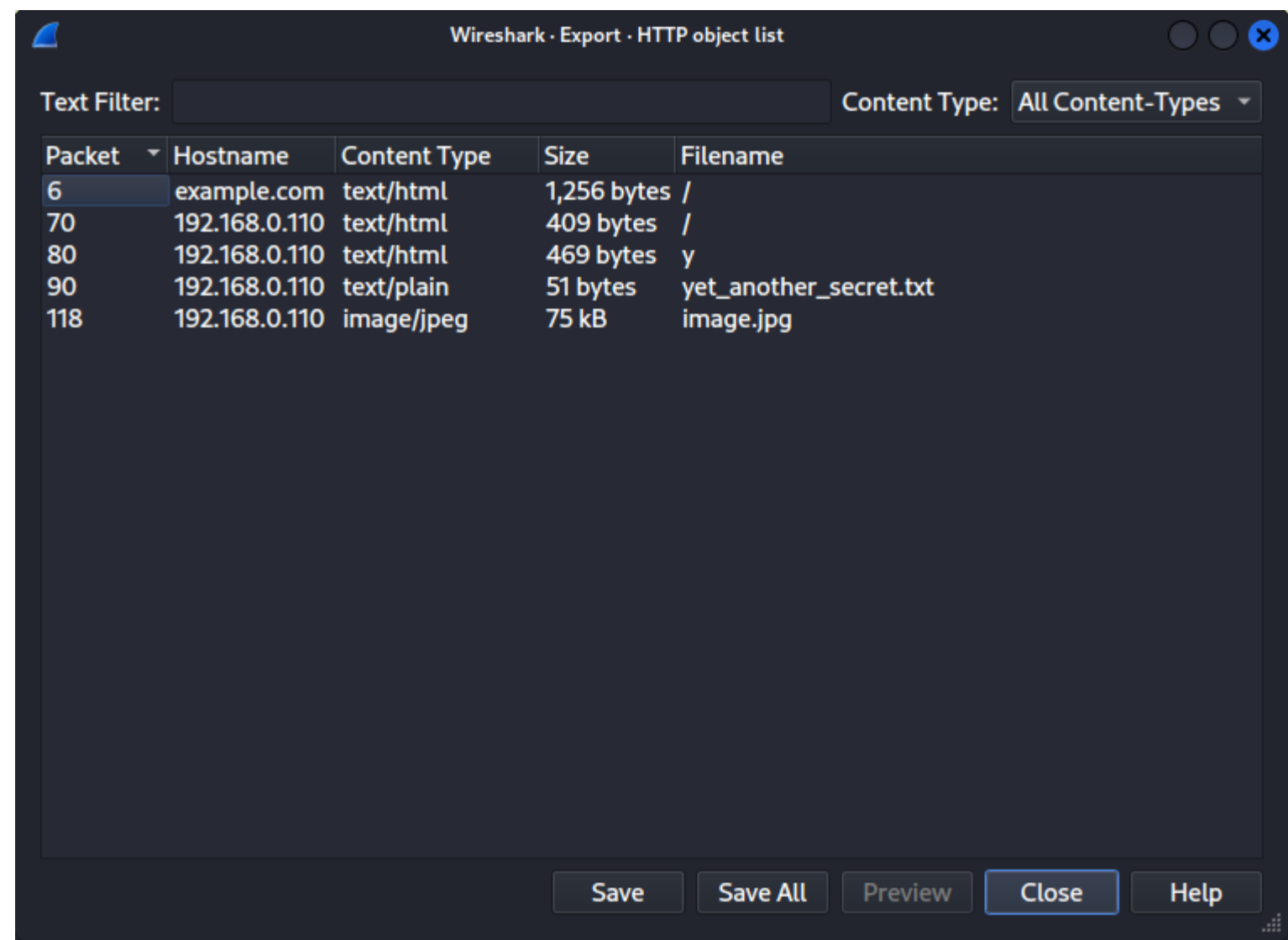
Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

Export Objects

The Export Objects feature allows us to extract files from captured network traffic. To access this feature, we can select File" → "Export Objects".

In the "HTTP" sub-menu, we can view a list of files that were transferred over HTTP during the capture. After selecting the file(s) that we want to extract, we can save them by clicking on the "Save" button.



Lab Questions

The organization that previously hired you to investigate the web attack has reached out to you again. This time, they have managed to capture the network traffic during the attack. They have provided you with the captured traffic file to help piece together the attacker's intentions and the extent of the damage. Your job is to analyze the captured traffic and answer the following questions:

1. What are the different protocols present in the captured traffic file?
2. It appears that the attacker is attempting to brute force the user's FTP password. Can you find any evidence of a correct password, and if so, what is it?
3. What additional information was the attacker able to extract from the user's FTP account?
4. What actions did the attacker take with the information obtained from the user's FTP account?
5. What's the root account password?
6. Can you identify the packet numbers in which the attacker exploited the Remote Code Execution vulnerability to gain access to the system? What was the exact payload used by the attacker?
7. After gaining access to the system, what does the attacker seem to be doing?
8. The attacker read a file from root's home directory. What was in that file?
9. The attacker downloaded a file inside root's home directory. What's the purpose of that file?
10. What information was transmitted through the attacker's covertly established channel of communication?

The traffic capture file can be downloaded from [challenge.pcapng](#).