

PENETRATION TESTING TOOLS

WIRESHARK & TCPDUMP

LLC 500 CYBERSECURITY AND ETHICAL HACKING

Linux Learning Centre Ltd
www.linuxlearningcentre.co.ke



Capture Files

When Wireshark loads, we are presented with a basic window where we can select the network interface we want to monitor as well as set display and capture filters.

02 we can use capture filters to reduce the amount of captured traffic by discarding any traffic that does not match our filter and narrow our focus to the packets we wish to analyze.

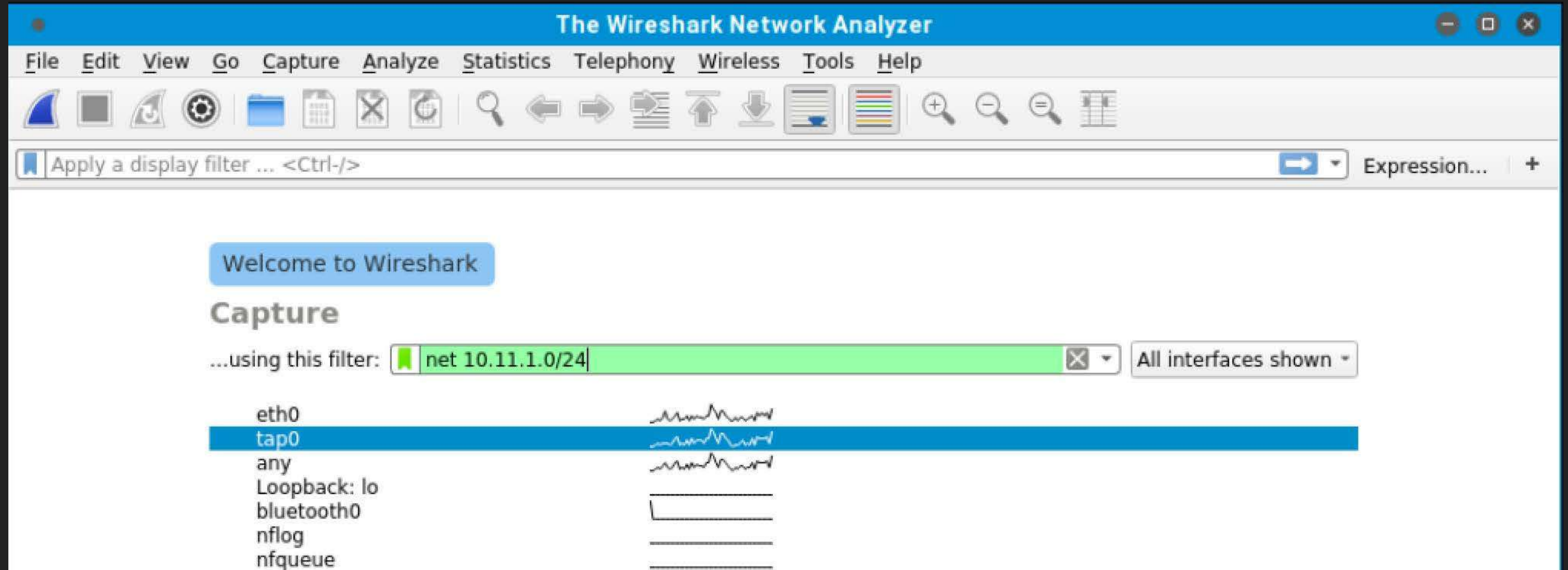
We'll start by selecting the interface we would like to monitor and entering a capture filter. In this case, we use the net filter to only capture traffic on the 10.11.1.0/24 address range:

It is also possible to choose from predefined capture filters by navigating to Capture > Capture filters, and we can also add our own capture filters by clicking on the + sign. With the capture filter set, we can start the capture by double-clicking our network interface (tap0) from the list of available interfaces.



Capture Files

02





Display Filters

03

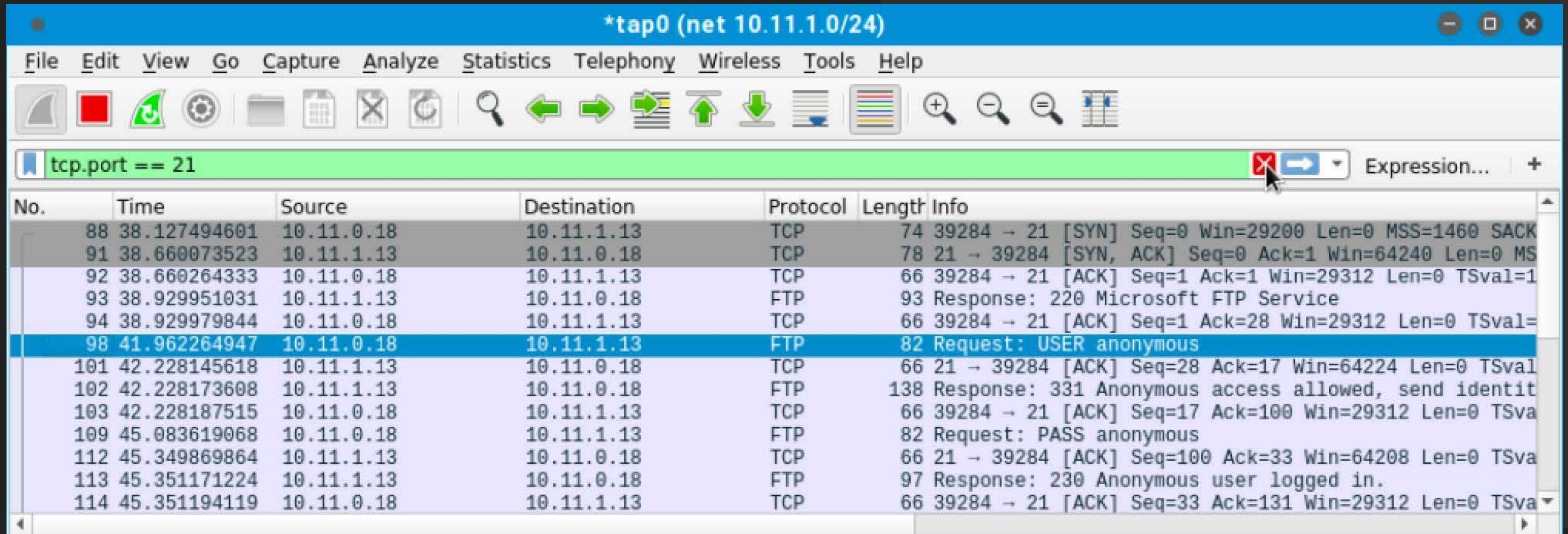
Now that Wireshark is capturing all the traffic on our local network, we can log in to an FTP server and inspect the traffic:

ftp 10.11.1.13



Let's apply a display filter that will only display FTP data, or TCP traffic on port 21

04



The screenshot shows the Wireshark network protocol analyzer interface. The title bar indicates the capture is on the interface `*tap0 (net 10.11.1.0/24)`. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for packet capture and analysis. The display filter bar at the top shows the filter `tcp.port == 21` in a green box, with a red 'X' icon and a blue arrow icon to its right, and a text field labeled 'Expression...'. Below the filter bar is a table of captured packets. The table has columns for No., Time, Source, Destination, Protocol, Length, and Info. The packets are filtered to show only those related to port 21, including TCP SYN, ACK, and FTP data.

No.	Time	Source	Destination	Protocol	Length	Info
88	38.127494601	10.11.0.18	10.11.1.13	TCP	74	39284 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK
91	38.660073523	10.11.1.13	10.11.0.18	TCP	78	21 → 39284 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MS
92	38.660264333	10.11.0.18	10.11.1.13	TCP	66	39284 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1
93	38.929951031	10.11.1.13	10.11.0.18	FTP	93	Response: 220 Microsoft FTP Service
94	38.929979844	10.11.0.18	10.11.1.13	TCP	66	39284 → 21 [ACK] Seq=1 Ack=28 Win=29312 Len=0 TSval=
98	41.962264947	10.11.0.18	10.11.1.13	FTP	82	Request: USER anonymous
101	42.228145618	10.11.1.13	10.11.0.18	TCP	66	21 → 39284 [ACK] Seq=28 Ack=17 Win=64224 Len=0 TSval
102	42.228173608	10.11.1.13	10.11.0.18	FTP	138	Response: 331 Anonymous access allowed, send identit
103	42.228187515	10.11.0.18	10.11.1.13	TCP	66	39284 → 21 [ACK] Seq=17 Ack=100 Win=29312 Len=0 TSva
109	45.083619068	10.11.0.18	10.11.1.13	FTP	82	Request: PASS anonymous
112	45.349869864	10.11.1.13	10.11.0.18	TCP	66	21 → 39284 [ACK] Seq=100 Ack=33 Win=64208 Len=0 TSva
113	45.351171224	10.11.1.13	10.11.0.18	FTP	97	Response: 230 Anonymous user logged in.
114	45.351194119	10.11.0.18	10.11.1.13	TCP	66	39284 → 21 [ACK] Seq=33 Ack=131 Win=29312 Len=0 TSva



Following TCP Streams



To view a particular TCP stream, we can right-click a packet of interest, such as the one containing the USER command in our FTP session, then select Follow > TCP Stream:

05

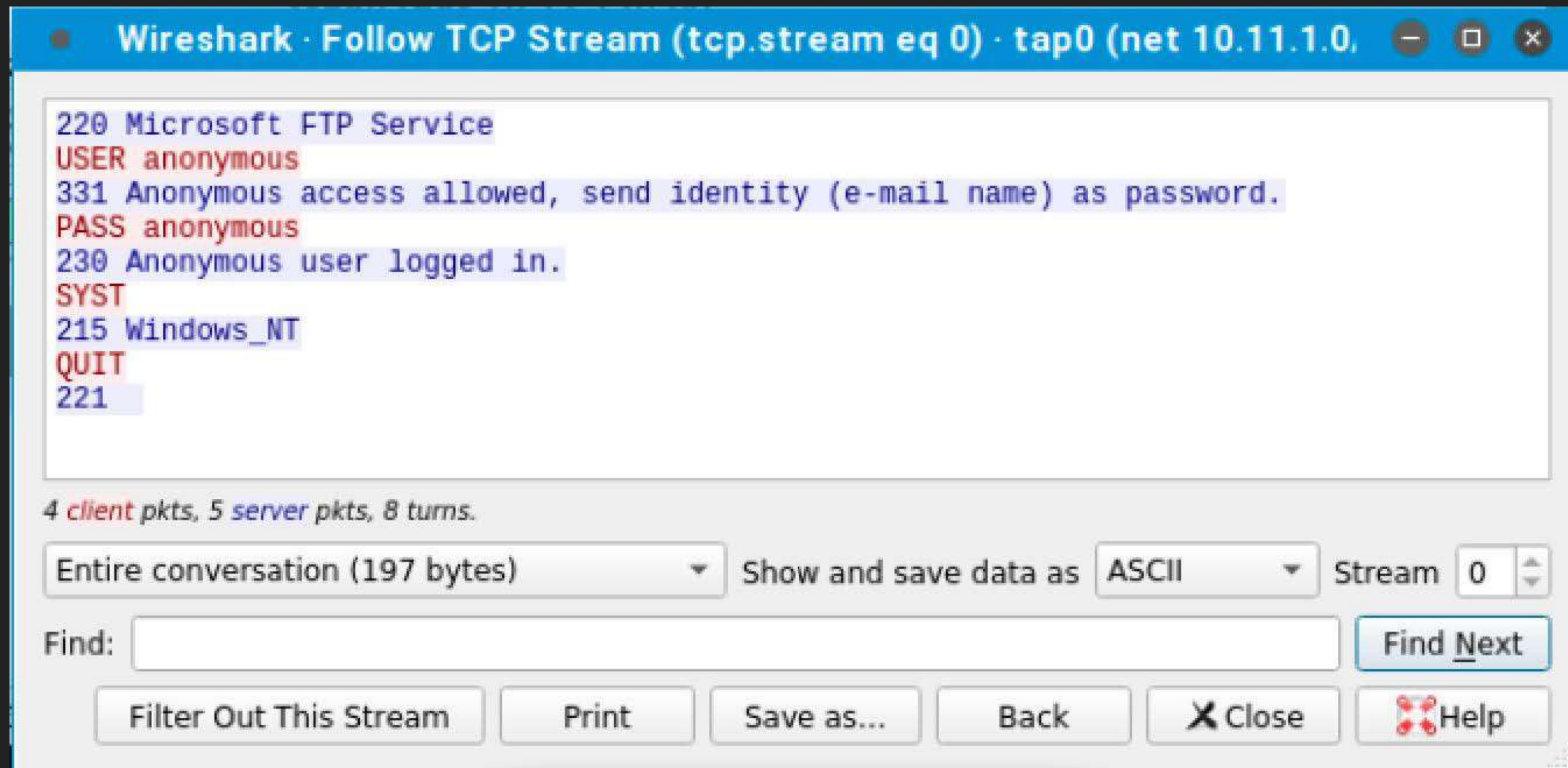
The screenshot shows the Wireshark interface with a packet capture filter 'tcp.port == 21'. The packet list shows several packets, with packet 98 selected. The packet details pane shows the structure of the selected packet: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and File Transfer Protocol (FTP). The packet bytes pane shows the raw data. The right-click context menu is open, and the 'Follow' option is selected, leading to a submenu where 'TCP Stream' is highlighted.

No.	Time	Source	Destination	Protocol	Length	Info
88	38.127494601	10.11.0.18	10.11.1.13	TCP	74	39284 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK
91	38.660073523	10.11.1.13	10.11.0.18	TCP	78	21 → 39284 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MS
92	38.660264333	10.11.0.18	10.11.1.13	TCP	66	39284 → 21 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1
93	38.929951031	10.11.1.13	10.11.0.18	FTP	93	Response: 220 Microsoft FTP Service
94	38.929979844	10.11.0.18	10.11.1.13	TCP	66	39284 → 21 [ACK] Seq=1 Ack=28 Win=29312 Len=0 TSval=
98	41.962264947	10.11.0.18	10.11.1.13	FTP	82	Request: USER anonymous
101	42.228145618	10.11.1.13	10.11.0.18	TCP	66	21 → 39284 [ACK] Seq=28 Ack=17 Win=64224 Len=0 TSval
102	42.228173608	10.11.1.13	10.11.0.18	FTP	138	Response: 331 Anonymous access allowed, send identit
103	42.228187515	10.11.1.13	10.11.0.18	TCP	66	39284 → 21 [ACK] Seq=17 Ack=100 Win=29312 Len=0 TSva
109	45.083619068	10.11.1.13	10.11.0.18	FTP	82	Request: PASS anonymous
112	45.349869864	10.11.1.13	10.11.0.18	TCP	66	21 → 39284 [ACK] Seq=100 Ack=33 Win=64208 Len=0 TSva
113	45.351171224	10.11.1.13	10.11.0.18	FTP	97	Response: 230 Anonymous user logged in.
114	45.351194119	10.11.1.13	10.11.0.18	TCP	66	39284 → 21 [ACK] Seq=33 Ack=131 Win=29312 Len=0 TSva

Frame 98: 82 bytes on wire (656 bits) on interface 0
Ethernet II, Src: Vmware_89:7c:af (00:50:56:89:7c:af)
Internet Protocol Version 4, Src: 10.11.0.18, Dest: 10.11.1.13
Transmission Control Protocol, Src Port: 39284, Dest Port: 21, Seq: 1, Ack: 28, Len: 16
File Transfer Protocol (FTP), Seq: 1, Len: 0
[Current working directory]

Follow > TCP Stream

The reassembled TCP stream is much easier to read, and we can review our interaction with the FTP server. Because FTP is a clear-text protocol, we can see the commands and output sent and received by our FTP client:





07

TCPDUMP

Tcpdump is a text-based network sniffer and most commonly-used command-line packet analyzer and can be found on most Unix and Linux operating systems





tcpdump

07

If you don't specify which network interface you'd like to capture traffic from, It will continue "dumping" the captured traffic to your terminal until you interrupt the command

To interrupt, press Ctrl + c.



tcpdump -D

If you have more than one network interface, then it'll be best to specify which interface you're trying to capture traffic on, since tcpdump may not choose the one you want by default. Use the -D option to print a list of network interfaces that tcpdump can use.

08

tcpdump -i enp0s3 -vv

We have a few different interfaces that we can use. Alternatively, we have the any option available that will let us capture traffic on all network interfaces simultaneously. If we want to capture network traffic on the enp0s3 interface, we would use the following command syntax. Use -v option to increase the verbosity



tcpdump -c 15

If you don't want tcpdump to endlessly output data to your terminal, you can use the -c option to specify how many packets you'd like the utility to capture. tcpdump will quit executing the command after the threshold has been reached, rather than waiting for you to interrupt. The following command will allow us to capture only the first 15 packets.

tcpdump -n

If you don't want tcpdump to perform DNS resolution on the network addresses in the output, you can use the -n option in your command. This will display all network addresses as IP addresses, rather than resolving them to domain names.

```
# tcpdump > traffic.txt
```

If you would rather save the network traffic output to file, instead of having it listed on your screen, you can always redirect the tcpdump output with the usual > and >> operators.

10

```
# tcpdump -n -w traffic.pcap
```

Another option is to write the network capture to file. These files usually have the .pcap file extension, and can't be read by an ordinary text editor.

```
# tcpdump -r traffic.pcap
```

To open the file for later analysis, use the -r option and the name of your file.

Interpret tcpdump command output

14:21:46.134249 IP 10.0.2.15.54000 > 104.16.168.35.443: Flags [.] , ack 2915, win 63000, length 0

11

Here's how to interpret that line of data:

- 14:21:46.134249 – Timestamp of when the packet was captured.
- IP 10.0.2.15.54000 – IP and port number of the source host.
- 104.16.168.35.443 – IP and port number of the destination host.
- Flags [.] – TCP flags (SYN, ACK, PSH, etc). [.] means ACK.
- ack 2915 – The acknowledgment number.
- win 63000 – The window number (bytes in receiving buffer).
- length 0 – The length of the payload data.



Filter tcpdump traffic

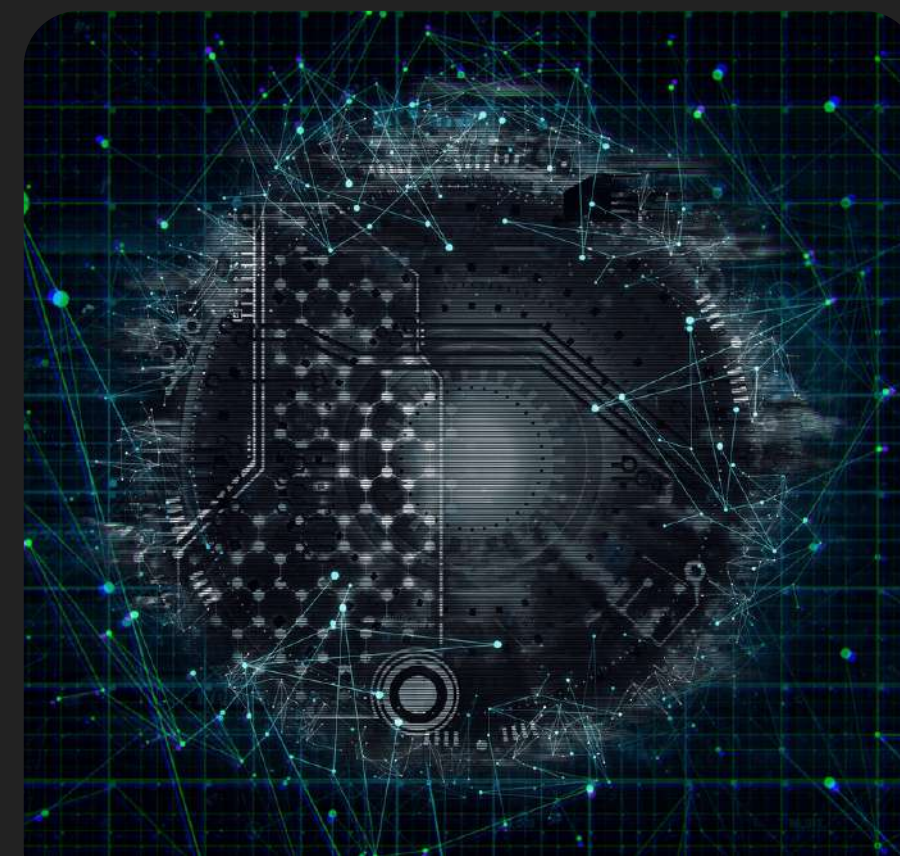
One of the best features of tcpdump is that we can filter out exactly the traffic we want to see. Without filtering out traffic by adapter, port number, and packet protocol, the amount of captured traffic can quickly become overwhelming and nearly impossible to sift through.



Despite the name tcpdump, we can use the tool to filter out all kinds of traffic, not just TCP.

For example, use the following syntax to filter out traffic that uses UDP.

```
# tcpdump -n udp
```





Filters

14

```
# tcpdump -n icmp
```

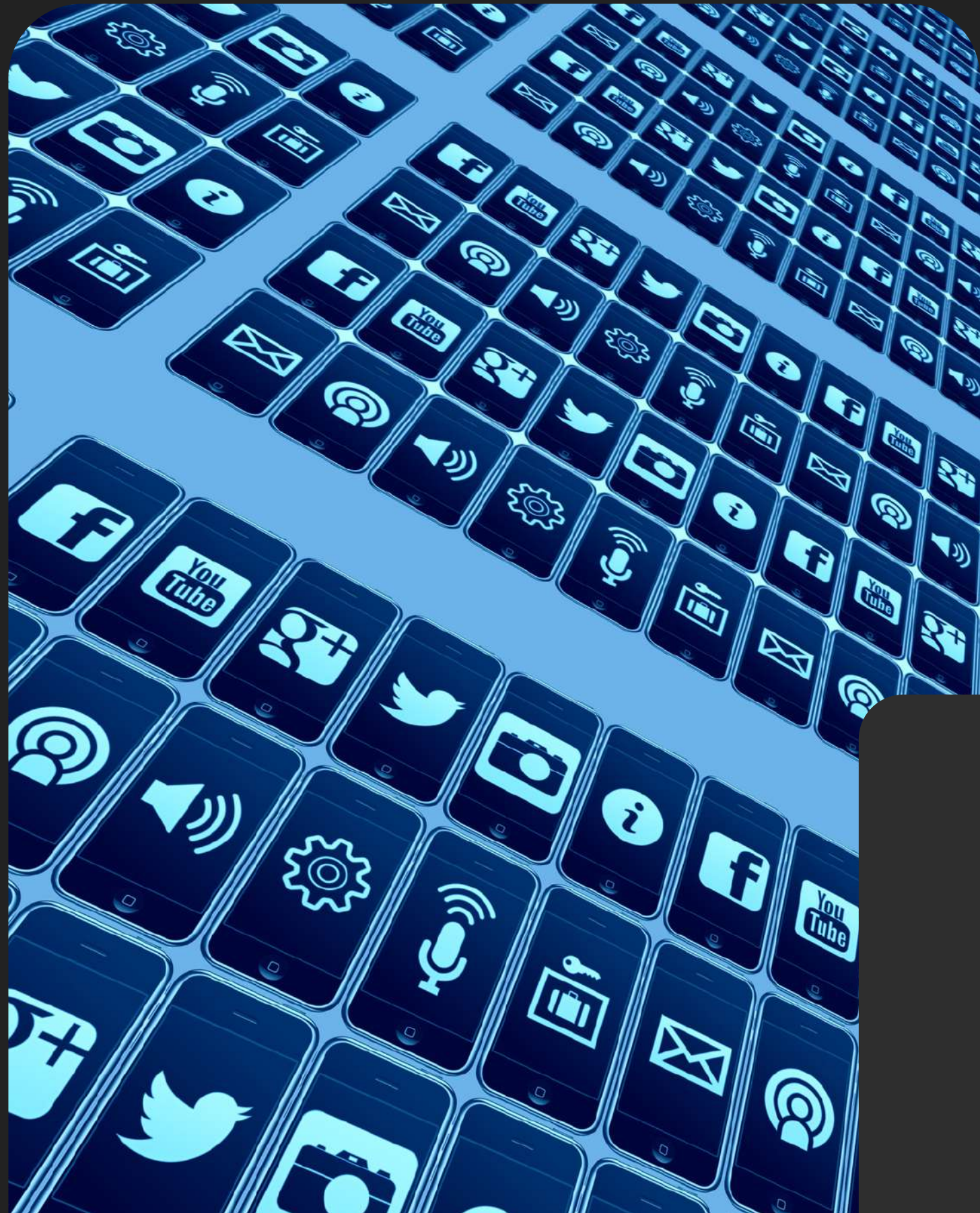
the following example that filters out ICMP:

```
# tcpdump -n proto 1
```

You can also use the corresponding protocol number to filter out a specific protocol. ICMP is protocol number 1



15



```
# tcpdump -n host 10.10.150.20
```

To filter traffic with a specific destination or source IP address, we can use the host qualifier with the -n option. For example, to filter traffic related to the host at IP address 10.10.150.20:





16

```
# tcpdump -n net 192.168.1
```

use the net qualifer if you want to filter out traffic to or from an entire network.

For example, the following command will filter traffic related to the 192.168.1.0/24 network.





tcpdump -n port 80

Use the port and portrange qualifiers to filter out packets related to a specific port or port range, respectively. For example, the following command will filter our traffic related to port 80 (HTTP)

tcpdump -n portrange 20-30

Or, to filter traffic from ports 20-30, the following command would be used.



tcpdump -n src host 10.10.150.20

Add the dst, src, src and dst, and src or dst qualifiers if you want to filter based on the source and/or destination address or port of the packets. For example, the following command will filter out packets that have a source IP address of 10.10.150.20.

tcpdump -n dst port 22

Or in this example, we filter out packets that are destined for the SSH port (port 22).



Combining filters

14

```
# tcpdump -n dst host 10.10.150.20 and tcp port 80
```

command will capture traffic that's destined for 10.10.150.20 on port 80 (HTTP).

```
# tcpdump -n 'dst host 10.10.150.20 and (tcp port 80 or tcp port 443)'
```

command will do the same as the previous, but also capture port 443 (HTTPS).



25

Wireshark & Tcpdump .

Cybersecurity and Ethical Hacking

