Wireshark - The Basics

Tool Overview

Use Cases

Wireshark is one of the most potent traffic analyser tools available in the wild. There are multiple purposes for its use:

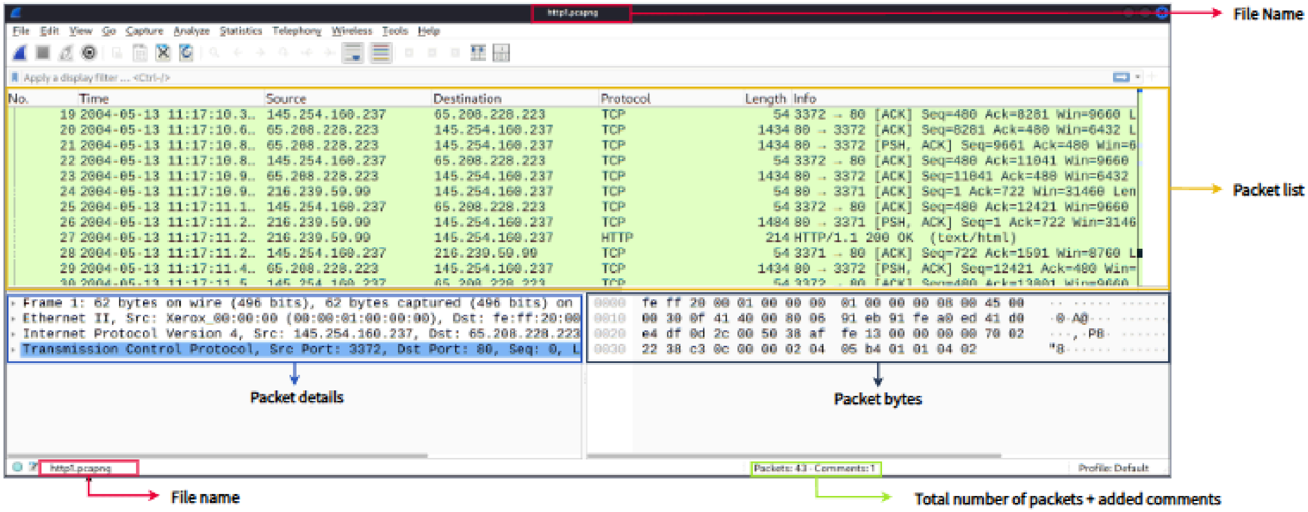
- Detecting and troubleshooting network problems, such as network load failure points and congestion.
- Detecting security anomalies, such as rogue hosts, abnormal port usage, and suspicious traffic.
- Investigating and learning protocol details, such as response codes and payload data.

GUI and Data

Wireshark GUI opens with a single all-in-one page, which helps users investigate the traffic in multiple ways. At first glance, five sections stand out.

Toolbar	The main toolbar contains multiple menus and shortcuts for packet sniffing and processing, including filtering, sorting, summarising, exporting and merging.
Display Filter Bar	The main query and filtering section.
Recent Files	List of the recently investigated files. You can recall listed files with a double-click.
Capture Filter and Interfaces	Capture filters and available sniffing points (network interfaces). The network interface is the connection point between a computer and a network. The software connection (e.g., lo, eth0 and ens33) enables networking hardware.
Status Bar	Tool status, profile and numeric packet information.

Loading PCAP Files


http1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
File Name
Packet list
Packet details
Packet bytes
http1.pcapng
File name
Packets: 43 · Comments: 1
Profile: Default
Total number of packets + added comments

```
No.   Time                Source           Destination      Protocol  Length Info
   19 2004-05-13 11:17:10.3.. 145.254.160.237  65.208.228.223   TCP       54 3372 → 80 [ACK] Seq=480 Ack=8281 Win=9660 L
   20 2004-05-13 11:17:10.6.. 65.208.228.223   145.254.160.237  TCP     1434 80 → 3372 [ACK] Seq=8281 Ack=480 Win=6432 L
   21 2004-05-13 11:17:10.8.. 65.208.228.223   145.254.160.237  TCP     1434 80 → 3372 [PSH, ACK] Seq=9661 Ack=480 Win=6
   22 2004-05-13 11:17:10.8.. 145.254.160.237  65.208.228.223   TCP       54 3372 → 80 [ACK] Seq=480 Ack=11041 Win=9660
   23 2004-05-13 11:17:10.9.. 65.208.228.223   145.254.160.237  TCP     1434 80 → 3372 [ACK] Seq=11041 Ack=480 Win=6432
   24 2004-05-13 11:17:10.9.. 216.239.59.99    145.254.160.237  TCP       54 80 → 3371 [ACK] Seq=1 Ack=722 Win=31460 Len
   25 2004-05-13 11:17:11.1.. 145.254.160.237  65.208.228.223   TCP       54 3372 → 80 [ACK] Seq=480 Ack=12421 Win=9660
   26 2004-05-13 11:17:11.2.. 216.239.59.99    145.254.160.237  TCP     1484 80 → 3371 [PSH, ACK] Seq=1 Ack=722 Win=3146
   27 2004-05-13 11:17:11.2.. 216.239.59.99    145.254.160.237  HTTP     214 HTTP/1.1 200 OK  (text/html)
   28 2004-05-13 11:17:11.2.. 145.254.160.237  216.239.59.99    TCP       54 3371 → 80 [ACK] Seq=722 Ack=1591 Win=8760 L
   29 2004-05-13 11:17:11.4.. 65.208.228.223   145.254.160.237  TCP     1434 80 → 3372 [PSH, ACK] Seq=12421 Ack=480 Win=
   30 2004-05-13 11:17:11.5.. 145.254.160.237  65.208.228.223   TCP       54 3372 → 80 [ACK] Seq=480 Ack=13801 Win=9660
```

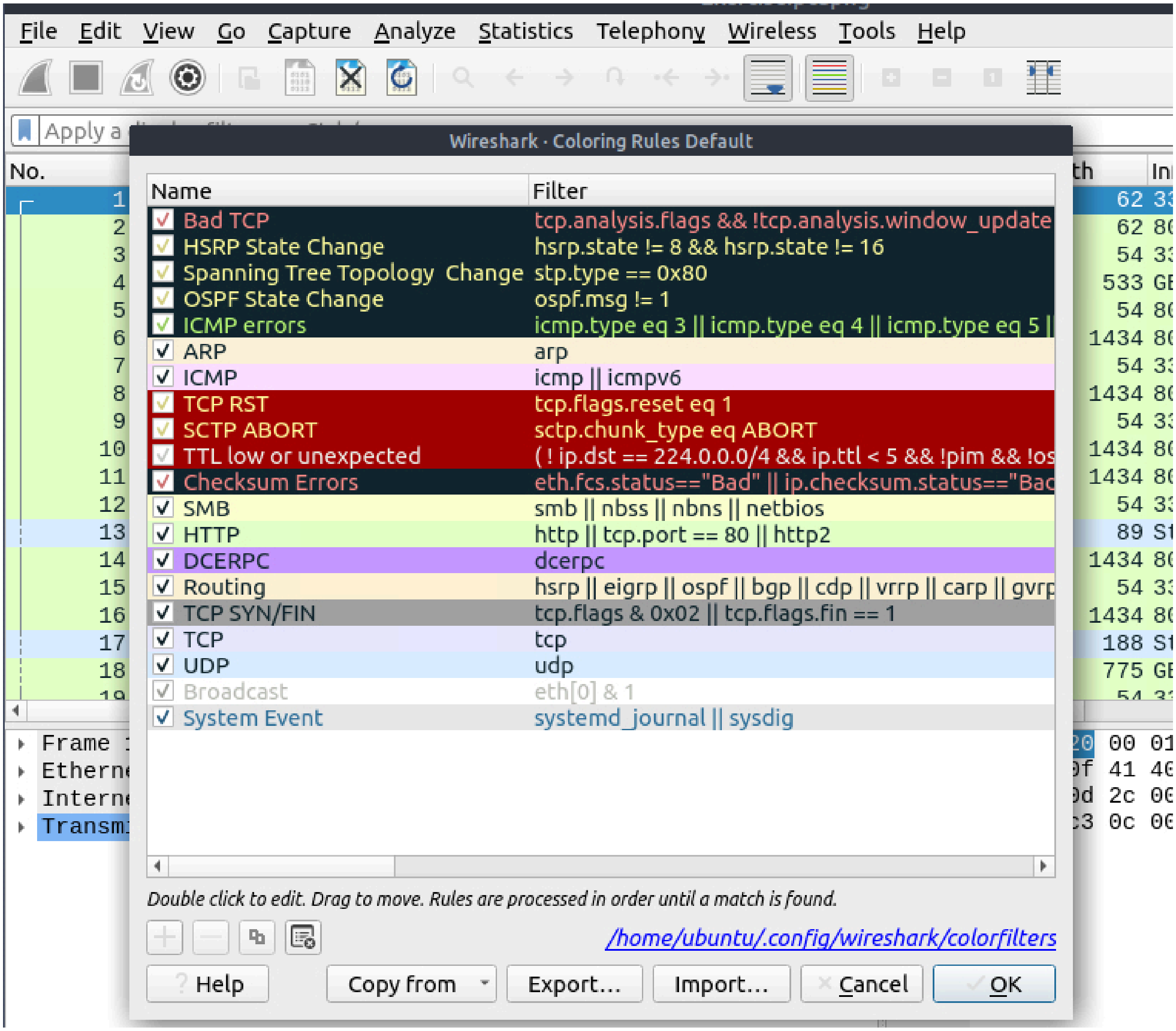
```
› Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on
› Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00
› Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
› Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, L
```

```
0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00   ·· ····· ······
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed 41 d0   ·0·A@·· ······
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00 70 02   ···,·P8· ······
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02         "8······ ······
```

Packet List Pane	Summary of each packet (source and destination addresses, protocol, and packet info). You can click on the list to choose a packet for further investigation. Once you select a packet, the details will appear in the other panels.
Packet Details Panel	Detailed protocol breakdown of the selected packet.
Packet Bytes Pane	Hex and decoded ASCII representation of the selected packet. It highlights the packet field depending on the clicked section in the details pane.

Colouring Packets

found under view > colouring rules



<table>
<tr><th>Name</th><th>Filter</th></tr>
<tr><td>☑ Bad TCP</td><td>tcp.analysis.flags && !tcp.analysis.window_update</td></tr>
<tr><td>☑ HSRP State Change</td><td>hsrp.state != 8 && hsrp.state != 16</td></tr>
<tr><td>☑ Spanning Tree Topology Change</td><td>stp.type == 0x80</td></tr>
<tr><td>☑ OSPF State Change</td><td>ospf.msg != 1</td></tr>
<tr><td>☑ ICMP errors</td><td>icmp.type eq 3 || icmp.type eq 4 || icmp.type eq 5 ||</td></tr>
<tr><td>☑ ARP</td><td>arp</td></tr>
<tr><td>☑ ICMP</td><td>icmp || icmpv6</td></tr>
<tr><td>☑ TCP RST</td><td>tcp.flags.reset eq 1</td></tr>
<tr><td>☑ SCTP ABORT</td><td>sctp.chunk_type eq ABORT</td></tr>
<tr><td>☑ TTL low or unexpected</td><td>(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim && !os</td></tr>
<tr><td>☑ Checksum Errors</td><td>eth.fcs.status=="Bad" || ip.checksum.status=="Bac</td></tr>
<tr><td>☑ SMB</td><td>smb || nbss || nbns || netbios</td></tr>
<tr><td>☑ HTTP</td><td>http || tcp.port == 80 || http2</td></tr>
<tr><td>☑ DCERPC</td><td>dcerpc</td></tr>
<tr><td>☑ Routing</td><td>hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvrp</td></tr>
<tr><td>☑ TCP SYN/FIN</td><td>tcp.flags & 0x02 || tcp.flags.fin == 1</td></tr>
<tr><td>☑ TCP</td><td>tcp</td></tr>
<tr><td>☑ UDP</td><td>udp</td></tr>
<tr><td>☑ Broadcast</td><td>eth[0] & 1</td></tr>
<tr><td>☑ System Event</td><td>systemd_journal || sysdig</td></tr>
</table>

Double click to edit. Drag to move. Rules are processed in order until a match is found.

/home/ubuntu/.config/wireshark/colorfilters

colour packets show different conditions and the protocol to spot anomalies and protocols in captures quickly.

Wireshark has two types of packet colouring methods: temporary rules that are only available during a program session and permanent rules that are saved under the preference file

You can use the "right-click menu" or **"View --> Coloring Rules"** menu to create permanent colouring rules. The **"Colourise Packet List"** menu activates/deactivates the colouring rules. Temporary packet colouring is done with the "right-click menu" or **"View --> Conversation Filter"** menu

Traffic Sniffing

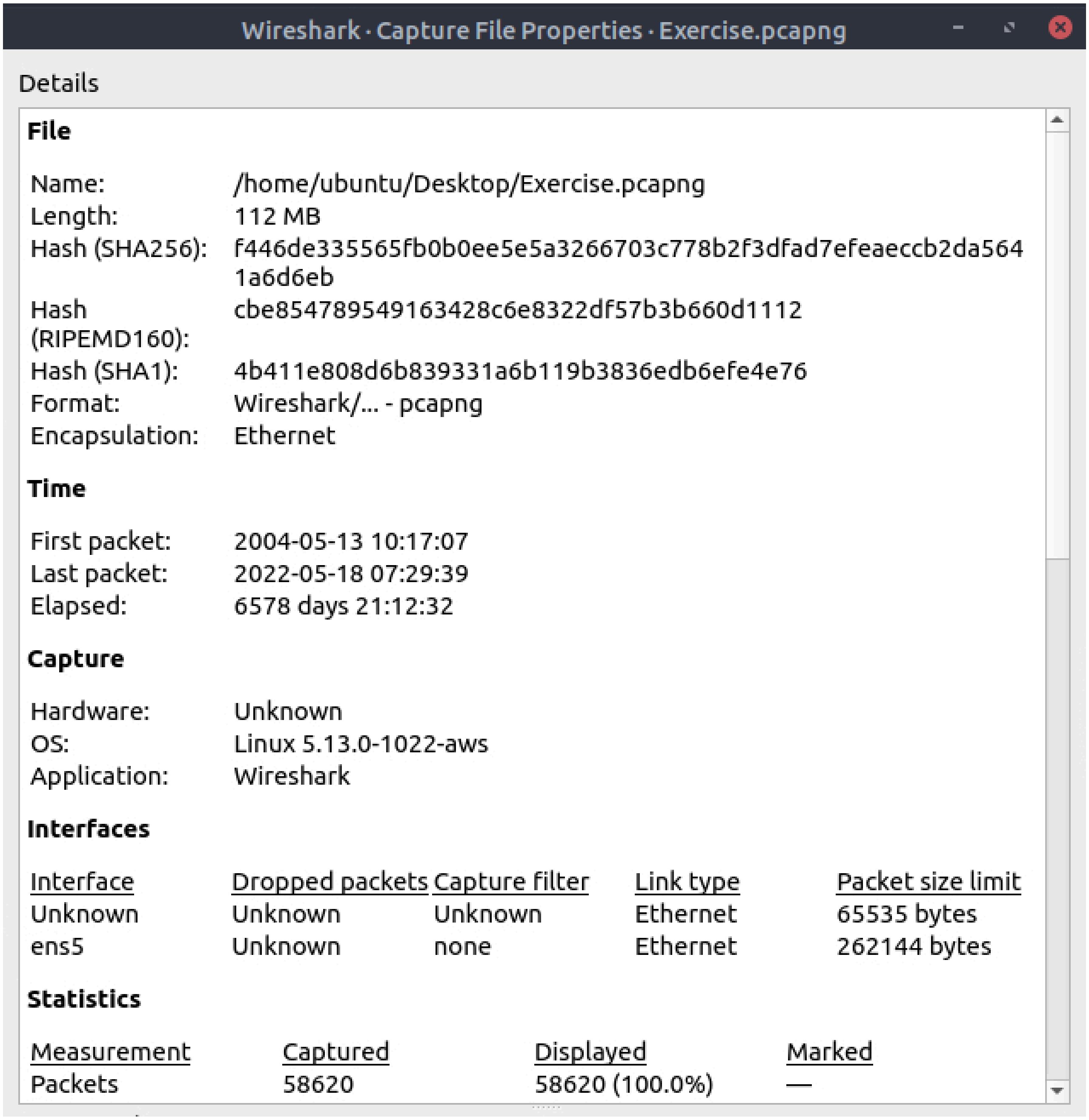
the blue **"shark button"** to start network sniffing (capturing traffic), the red button will stop the sniffing, and the green button will restart the sniffing process.

Merge PCAP Files

You can use the **"File --> Merge"** menu path to merge a pcap with the processed one. When you choose the second file, Wireshark will show the total number of packets in the selected file. Once you click "open", it will merge the existing pcap file with the chosen one and create a new pcap file.

View File Details

Knowing the file details is helpful. Especially when working with multiple pcap files, sometimes you will need to know and recall the file details (File hash, capture time, capture file comments, interface and statistics) to identify the file, classify and prioritise it. You can view the details by following "**Statistics --> Capture File Properties"** or by clicking the **"pcap icon located on the left bottom"**.



Wireshark · Capture File Properties · Exercise.pcapng

Details

File

Name: /home/ubuntu/Desktop/Exercise.pcapng
Length: 112 MB
Hash (SHA256): f446de335565fb0b0ee5e5a3266703c778b2f3dfad7efeaeccb2da5641a6d6eb
Hash (RIPEMD160): cbe854789549163428c6e8322df57b3b660d1112
Hash (SHA1): 4b411e808d6b839331a6b119b3836edb6efe4e76
Format: Wireshark/... - pcapng
Encapsulation: Ethernet

Time

First packet: 2004-05-13 10:17:07
Last packet: 2022-05-18 07:29:39
Elapsed: 6578 days 21:12:32

Capture

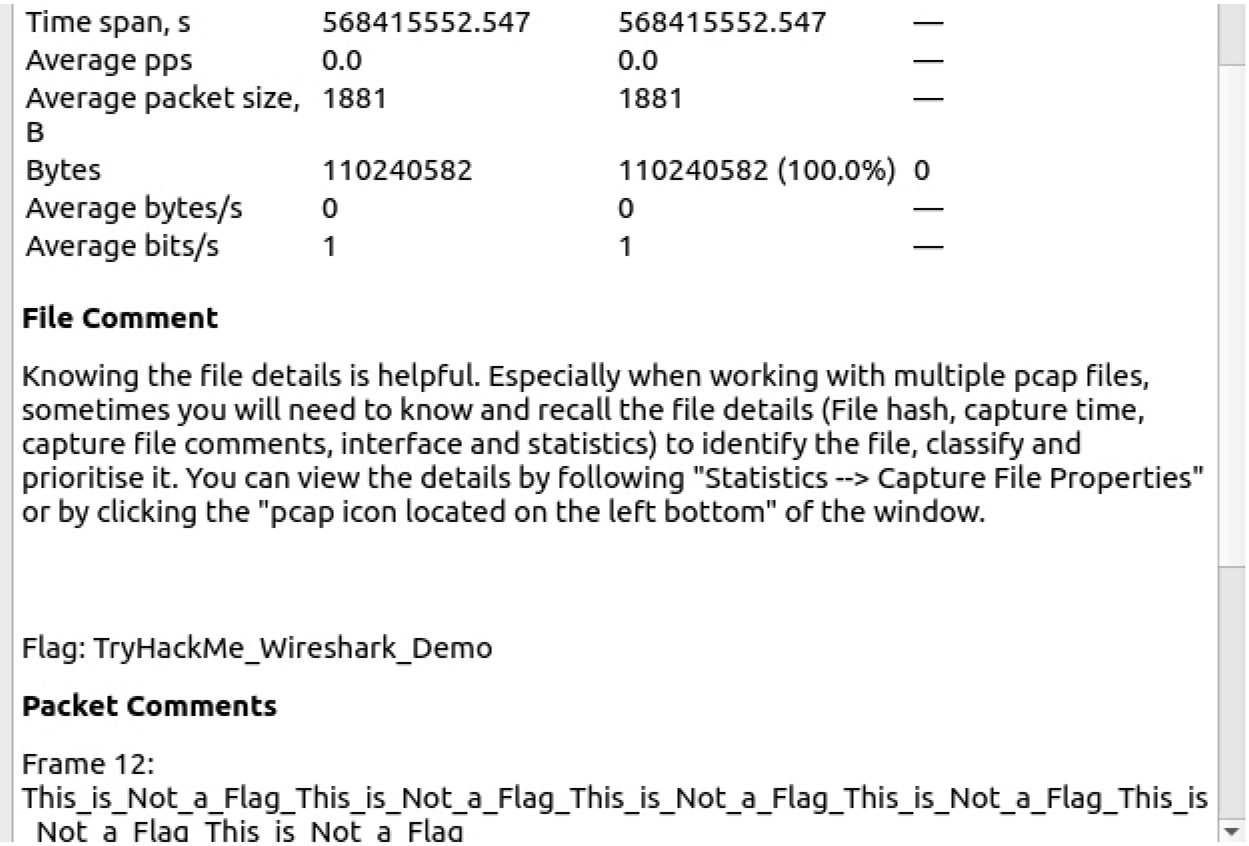
Hardware: Unknown
OS: Linux 5.13.0-1022-aws
Application: Wireshark

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	65535 bytes
ens5	Unknown	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	58620	58620 (100.0%)	—



Time span, s	568415552.547	568415552.547	—
Average pps	0.0	0.0	—
Average packet size, B	1881	1881	—
Bytes	110240582	110240582 (100.0%)	0
Average bytes/s	0	0	—
Average bits/s	1	1	—

File Comment

Knowing the file details is helpful. Especially when working with multiple pcap files, sometimes you will need to know and recall the file details (File hash, capture time, capture file comments, interface and statistics) to identify the file, classify and prioritise it. You can view the details by following "Statistics --> Capture File Properties" or by clicking the "pcap icon located on the left bottom" of the window.

Flag: TryHackMe_Wireshark_Demo

Packet Comments

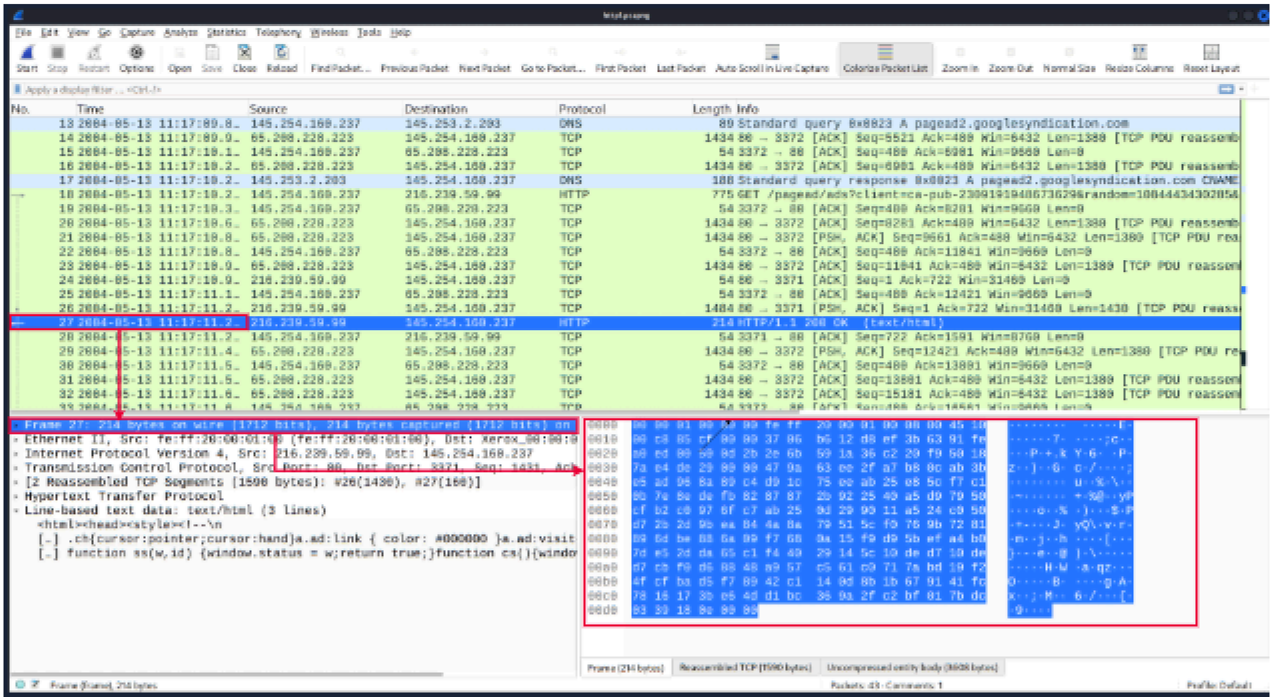
Frame 12:
This_is_Not_a_Flag_This_is_Not_a_Flag_This_is_Not_a_Flag_This_is_Not_a_Flag_This_is_Not_a_Flag_This_is_Not_a_Flag

Capture file properties under the statistics tab can tell us alot about the pcapng file and i managed to find a flag in here.

Packet Dissection

also known as protocol dissection which investigates packet details by decoding available protocols and fields

Packets consist of 5 to 7 layers based on the OSI model.



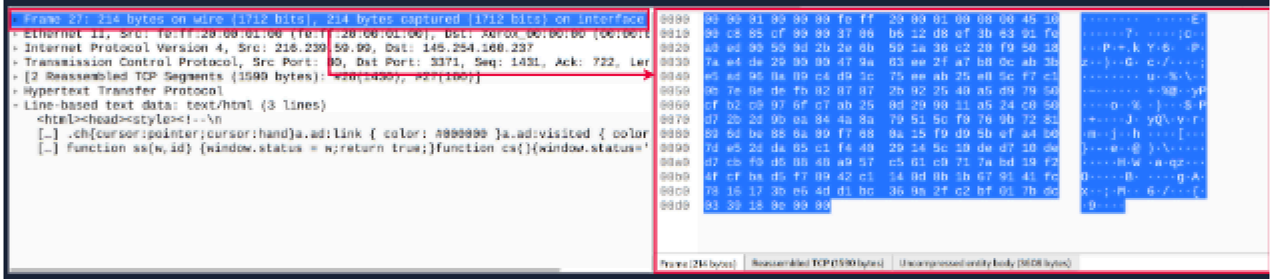
```
No.  Time                       Source           Destination      Protocol  Length Info
  13 2004-05-13 11:17:09.8…  145.254.160.237  145.253.2.203    DNS         89 Standard query 0x0023 A pagead2.googlesyndication.com
  14 2004-05-13 11:17:09.9…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [ACK] Seq=5521 Ack=480 Win=6432 Len=1380 [TCP PDU reassemb
  15 2004-05-13 11:17:10.1…  145.254.160.237  65.208.228.223   TCP         54 3372 → 80 [ACK] Seq=480 Ack=6901 Win=9660 Len=0
  16 2004-05-13 11:17:10.2…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [ACK] Seq=6901 Ack=480 Win=6432 Len=1380 [TCP PDU reassemb
  17 2004-05-13 11:17:10.2…  145.253.2.203    145.254.160.237  DNS        188 Standard query response 0x0023 A pagead2.googlesyndication.com CNAME
  18 2004-05-13 11:17:10.2…  145.254.160.237  216.239.59.99    HTTP       775 GET /pagead/ads?client=ca-pub-2309191948673629&random=10844434302854
  19 2004-05-13 11:17:10.3…  145.254.160.237  65.208.228.223   TCP         54 3372 → 80 [ACK] Seq=480 Ack=8281 Win=9660 Len=0
  20 2004-05-13 11:17:10.6…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [ACK] Seq=8281 Ack=480 Win=6432 Len=1380 [TCP PDU reassemb
  21 2004-05-13 11:17:10.8…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [PSH, ACK] Seq=9661 Ack=480 Win=6432 Len=1380 [TCP PDU rea
  22 2004-05-13 11:17:10.8…  145.254.160.237  65.208.228.223   TCP         54 3372 → 80 [ACK] Seq=480 Ack=11041 Win=9660 Len=0
  23 2004-05-13 11:17:10.9…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [ACK] Seq=11041 Ack=480 Win=6432 Len=1380 [TCP PDU reassem
  24 2004-05-13 11:17:10.9…  216.239.59.99    145.254.160.237  TCP         54 80 → 3371 [ACK] Seq=1 Ack=722 Win=31460 Len=0
  25 2004-05-13 11:17:11.1…  145.254.160.237  65.208.228.223   TCP         54 3372 → 80 [ACK] Seq=480 Ack=12421 Win=9660 Len=0
  26 2004-05-13 11:17:11.2…  216.239.59.99    145.254.160.237  TCP       1484 80 → 3371 [PSH, ACK] Seq=1 Ack=722 Win=31460 Len=1430 [TCP PDU reass
  27 2004-05-13 11:17:11.2…  216.239.59.99    145.254.160.237  HTTP       214 HTTP/1.1 200 OK  (text/html)
  28 2004-05-13 11:17:11.2…  145.254.160.237  216.239.59.99    TCP         54 3371 → 80 [ACK] Seq=722 Ack=1591 Win=8760 Len=0
  29 2004-05-13 11:17:11.4…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [PSH, ACK] Seq=12421 Ack=480 Win=6432 Len=1380 [TCP PDU re
  30 2004-05-13 11:17:11.5…  145.254.160.237  65.208.228.223   TCP         54 3372 → 80 [ACK] Seq=480 Ack=13801 Win=9660 Len=0
  31 2004-05-13 11:17:11.5…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [ACK] Seq=13801 Ack=480 Win=6432 Len=1380 [TCP PDU reassem
  32 2004-05-13 11:17:11.6…  65.208.228.223   145.254.160.237  TCP       1434 80 → 3372 [ACK] Seq=15181 Ack=480 Win=6432 Len=1380 [TCP PDU reassem
  33 2004-05-13 11:17:11.8…  145.254.160.237  65.208.228.223   TCP         54 3372 → 80 [ACK] Seq=480 Ack=16561 Win=9660 Len=0
```

```
› Frame 27: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on
› Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00:00:0
› Internet Protocol Version 4, Src: 216.239.59.99, Dst: 145.254.160.237
› Transmission Control Protocol, Src Port: 80, Dst Port: 3371, Seq: 1431, Ack
› [2 Reassembled TCP Segments (1590 bytes): #26(1430), #27(160)]
› Hypertext Transfer Protocol
› Line-based text data: text/html (3 lines)
    <html><head><style><!--\n
    […] .ch{cursor:pointer;cursor:hand}a.ad:link { color: #000000 }a.ad:visit
    […] function ss(w,id) {window.status = w;return true;}function cs(){windo
```

```
0000  00 00 01 00 00 00 fe ff  20 00 01 00 08 00 45 10
0010  00 c8 05 cf 00 00 37 06  b6 12 d8 ef 3b 63 91 fe
0020  a0 ed 00 50 0d 2b 2e 6b  59 1a 36 c2 20 f9 50 18
0030  7a e4 de 29 00 00 47 9a  63 ee 2f a7 b8 0c ab 3b
0040  e5 ad 96 8a 89 c4 d9 1c  75 ee ab 25 e8 5c f7 c1
0050  0b 7e 8e de fb 82 87 87  2b 92 25 40 a5 d9 79 50
0060  cf b2 c0 97 6f c7 ab 25  0d 29 90 11 a5 24 c0 50
0070  d7 2b 2d 9b ea 84 4a 8a  79 51 5c f0 76 9b 72 81
0080  89 6d be 88 6a 89 f7 68  0a 15 f9 d9 5b ef a4 b0
0090  7d e5 2d da 65 c1 f4 40  29 14 5c 10 de d7 10 de
00a0  d7 cb f0 d6 88 48 a9 57  c5 61 c0 71 7a bd 19 f2
00b0  4f cf ba d5 f7 89 42 c1  14 0d 8b 1b 67 91 41 fc
00c0  78 16 17 3b e6 4d d1 bc  36 9a 2f c2 bf 01 7b dc
00d0  03 39 18 0e 00 00
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Start Stop Restart Options Open Save Close Reload Find Packet… Previous Packet Next Packet Go to Packet… First Packet Last Packet Auto Scroll in Live Capture Colorize Packet List Zoom In Zoom Out Normal Size Resize Columns Reset Layout
Apply a display filter … <Ctrl-/>
Frame (214 bytes) Reassembled TCP (1590 bytes) Uncompressed entity body (3608 bytes)
Frame (frame), 214 bytes
Packets: 43 · Comments: 1
Profile: Default

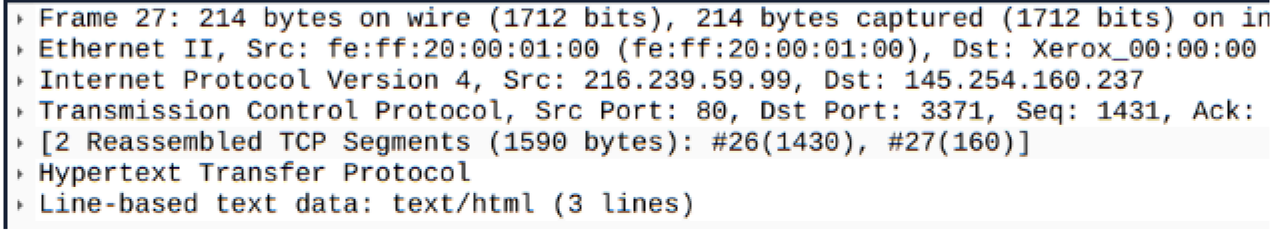
Each time you click a detail, it will highlight the corresponding part in the packet bytes pane.



```
› Frame 27: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface
› Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00:00:00 (00:00:01:
› Internet Protocol Version 4, Src: 216.239.59.99, Dst: 145.254.160.237
› Transmission Control Protocol, Src Port: 80, Dst Port: 3371, Seq: 1431, Ack: 722, Len
› [2 Reassembled TCP Segments (1590 bytes): #26(1430), #27(160)]
› Hypertext Transfer Protocol
› Line-based text data: text/html (3 lines)
    <html><head><style><!--\n
    […] .ch{cursor:pointer;cursor:hand}a.ad:visited { color: #000000 }a.ad:visited { color
    […] function ss(w,id) {window.status = w;return true;}function cs(){window.status='
```

Frame (214 bytes) Reassembled TCP (1590 bytes) Uncompressed entity body (3608 bytes)

Let's have a closer view of the details pane.



```
› Frame 27: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on in
› Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00:00:00
› Internet Protocol Version 4, Src: 216.239.59.99, Dst: 145.254.160.237
› Transmission Control Protocol, Src Port: 80, Dst Port: 3371, Seq: 1431, Ack:
› [2 Reassembled TCP Segments (1590 bytes): #26(1430), #27(160)]
› Hypertext Transfer Protocol
› Line-based text data: text/html (3 lines)
```

The Frame (Layer 1): This will show you what frame/packet you are looking at and details specific to the Physical layer of the OSI model.

Source [MAC] (Layer 2): This will show you the source and destination MAC Addresses; from the Data Link layer of the OSI model.

Source [IP] (Layer 3): This will show you the source and destination IPv4 Addresses; from the Network layer of the OSI model.

Protocol (Layer 4): This will show you details of the protocol used (UDP/TCP) and source and destination ports; from the Transport layer of the OSI model.

Protocol Errors: This continuation of the 4th layer shows specific segments from TCP that needed to be reassembled.

**Application Protocol (Layer 5): T his will show details specific to the protocol used, such as HTTP, FTP, and SMB. From the Application layer of the OSI model.

Application Data: This extension of the 5th layer can show the application-specific data.

Packet Navigation

Packet Numbers

helps analyse process for big captures and easy to go back to a specific point of an event.

Go to Packet

Packet numbers do not only help to count the total number of packets or make it easier to find/investigate specific packets.

provides in-frame packet tracking and finds the next packet in the particular part of the conversation

can use the **"Go"** menu and toolbar to view specific packets

Find Packets

Wireshark can find packets by packet content. You can use the **"Edit --> Find Packet"** menu to make a search inside the packets for a particular event of interest.

There are two crucial points in finding packets. The first is knowing the input type. This functionality accepts four types of inputs (Display filter, Hex, String and Regex). String and regex searches are the most commonly used search types

The second point is choosing the search field. You can conduct searches in the three panes (packet list, packet details, and packet bytes), and it is important to know the available information in each pane to find the event of interest.

Mark Packets

can find/point to a specific packet for further investigation by marking it

an use the **"Edit"** or the **"right-click"** menu to mark/unmark packets

Packet Comments

You can add comments for particular packets that will help the further investigation or remind and point out important/suspicious points for other layer analysts.

Export Packets

sometimes, it is necessary to separate specific packages from the file and dig deeper to resolve an incident. This functionality helps analysts share the only suspicious packages.
You can use the **"File"** menu to export packets.

Export Objects (Files)

Wireshark can extract files transferred through the wire. For a security analyst, it is vital to discover shared files and save them for further investigation. Exporting objects are available only for selected protocol's streams (DICOM, HTTP, IMF, SMB and TFTP).

Time Display Format

Wireshark shows the time in "Seconds Since Beginning of Capture", the common usage is using the UTC Time Display Format for a better view. You can use the **"View --> Time Display Format"** menu to change the time display format.

Expert Info

Wireshark also detects specific states of protocols to help analysts easily spot possible anomalies and problems. Note that these are only suggestions, and there is always a chance of having false positives/negatives.

Severity	**Colour**	**Info**
Chat	**Blue**	Information on usual workflow.
Note	**Cyan**	Notable events like application error codes.
Warn	**Yellow**	Warnings like unusual error codes or problem statements.
Error	**Red**	Problems like malformed packets.

Severity	**Colour**	**Info**
Chat	**Blue**	Information on usual workflow.
Note	**Cyan**	Notable events like application error codes.
Warn	**Yellow**	Warnings like unusual error codes or problem statements.

Error	**Red**	Problems like malformed packets.

You can use the **"lower left bottom section"** in the status bar or **"Analyse --> Expert Information"** menu to view all available information entries via a dialogue box. It will show the packet number, summary, group protocol and total occurrence.

can find md5sum of a packet if you export bytes on packet and name file save somewhere then in terminal find directory of file then do md5sum [file name]

Packet Filtering

Wireshark has two types of filtering approaches: capture and display filters. Capture filters are used for **"capturing"** only the packets valid for the used filter. Display filters are used for **"viewing"** the packets valid for the used filter.

Apply as Filter

can click on the field you want to filter and use the "right-click menu" or **"Analyse --> Apply as Filter"** menu to filter the specific value.

Conversation Filter

the "Conversation Filter" option helps you view only the related packets and hide the rest of the packets easily. You can use the"right-click menu" or "**Analyse --> Conversation Filter**" menu to filter conversations.

Colourise Conversation

This option is similar to the "Conversation Filter" with one difference. It highlights the linked packets without applying a display filter and decreasing the number of viewed packets. can use the "right-click menu" or **"View --> Colourise Conversation"** menu to colourise a linked packet in a single click. Note that you can use the **"View --> Colourise Conversation --> Reset Colourisation"** menu to undo this operation

Prepare as Filter

Similar to "Apply as Filter", this option helps analysts create display filters using the "right-click" menu. adds the required query to the pane and waits for the execution command (enter) or another chosen filtering option by using the **".. and/or.."** from the "right-click menu".

Apply as Column

You can use the "right-click menu" or "**Analyse --> Apply as Column**" menu to add columns to the packet list pane. Once you click on a value and apply it as a column, it will be visible on the packet list pane

This function helps analysts examine the appearance of a specific value/field across the available packets in the capture file

Follow Stream

Following the protocol, streams help analysts recreate the application-level data and understand the event of interest. It is also possible to view the unencrypted protocol data like usernames, passwords and other transferred data. Can use the"right-click menu" or **"Analyse --> Follow TCP/UDP/HTTP Stream"** menu to follow traffic streams.

Streams are shown in a separate dialogue box; packets originating from the server are highlighted with blue, and those originating from the client are highlighted with red.