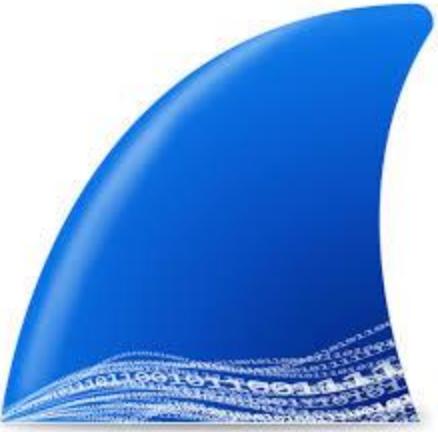


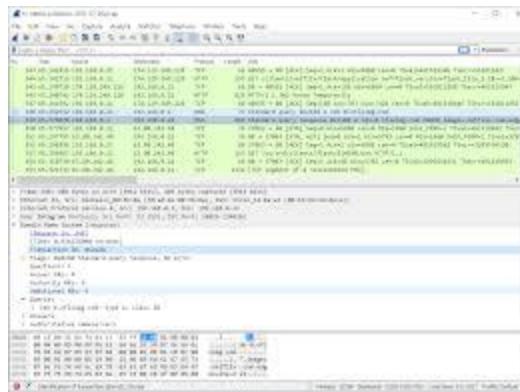


# Wireshark

# What is Wireshark?



Wireshark is a free,  
open-source network protocol  
analyser.



you capture and inspect  
network traffic in real time.  
Every single packet is  
captured



Think of it as a  
microscope for your  
network — zooming in  
on every packet!

# WHY ?



- Troubleshoot network issues.
- Spot suspicious or malicious traffic
- Debugging apps

Guys we did it we found the intruder



- Learn how protocols work.
- Debug network applications.
- Follow streams like conversations



- Finding slow networks.
- Detecting intrusions or malware traffic
- Analysing VoIP (record and play back phone calls or determine quality to fix)
- Learning networking

# HOW DOES IT WORK



Decodes hundreds of protocols (TCP, HTTP, DNS, etc.). By capturing packets live, Display & capture filters.



1. Captures (sniffs) packets from network interfaces.
2. Decodes protocol layers.
3. Applies filters to narrow results.
4. Displays packet metadata + payload.
5. Allows analysis, exporting, and reporting

# W AND L OF WIRESHARK

You get to see what devices are really saying behind the scenes.

Feels like hacking in a movie (but legal!).

You can follow streams like reading a conversation.

Spot weird traffic, misbehaving apps, or even memes being sent over HTTP.

Limitations:

- HTTPS sites are encrypted (SSL/TLS)
  - Can be unencrypted if you own the web server
  - Other protocols aren't – DNS packets show the websites people are accessing on the network you're connected to
- Requires admin permissions
- Can be overwhelming for beginners



Pros:

- Free, powerful, open-source
- Deep protocol support
- Great for learning

# Wireshark Filters/Tips

Play around with filters — it's like magic!

# Filter

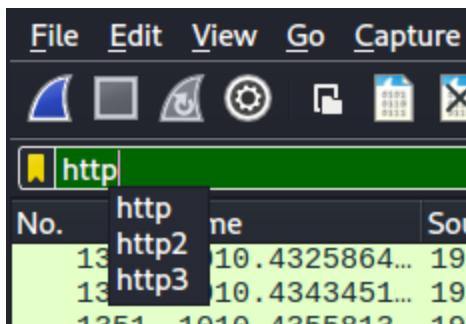
http

DNS

tcp.port == 80

ip.addr == 8.8.8.8

tcp.flags.syn == 1



- Use capture filters to reduce noise.
  - Save large captures in segments.
  - Use colour rules to highlight patterns.
  - Learn display filters—they’re powerful!
  - Be mindful of sensitive captured data.



# Interface

| No.  | Time        | Source      | Destination  | Protocol | Length | Info                   |
|------|-------------|-------------|--------------|----------|--------|------------------------|
| 2731 | 8.625194349 | 2.19.252.91 | 192.168.60.1 | TCP      | 1514   | 443 → 57369 [PSH, ACK] |
| 2732 | 8.625194373 | 2.19.252.91 | 192.168.60.1 | TCP      | 1514   | 443 → 57369 [PSH, ACK] |

Column Description

No. The packet number in the capture sequence.

Time The time elapsed since the start of the capture.

Source The IP or MAC address from which the packet originated.

Destination The IP or MAC address to which the packet is being sent.

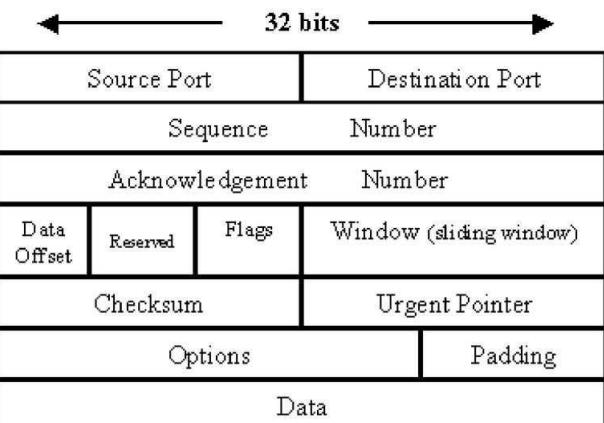
Protocol The protocol type, for example, TLS v1.2, TCP, ICMP, etc.

Length The size of the packet in bytes.

Info A summary of the packet's function or contents.

Header information/meta data of packet

## TCP Packet header



wireshark.pcapng

File Analyze Statistics Telephony Wireless Tools Help

Frame 184: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) (0% loss).

Ethernet II, Src: PCSSystem (52:54:00:12:35:02) [ether], Dst: PCSSystem (08:00:27:58:6e:af) [ether]

Internet Protocol Version 4, Src: 185.125.190.96, Dst: 10.0.2.15

Transmission Control Protocol, Src Port: 80, Dst Port: 34006, Seq: 186, Ack: 187, Len: 60

Source Port: 80  
Destination Port: 34006  
[Stream index: 20]  
[Conversation completeness: Complete, WITH\_DATA (31)]  
[TCP Segment Len: 0]  
Sequence Number: 186 (relative sequence number)  
Sequence Number (raw): 79872187  
[Next Sequence Number: 187 (relative sequence number)]  
Acknowledgment Number: 189 (relative ack number)  
Acknowledgment number (raw): 700296643  
0101... = Header Length: 20 bytes (5)

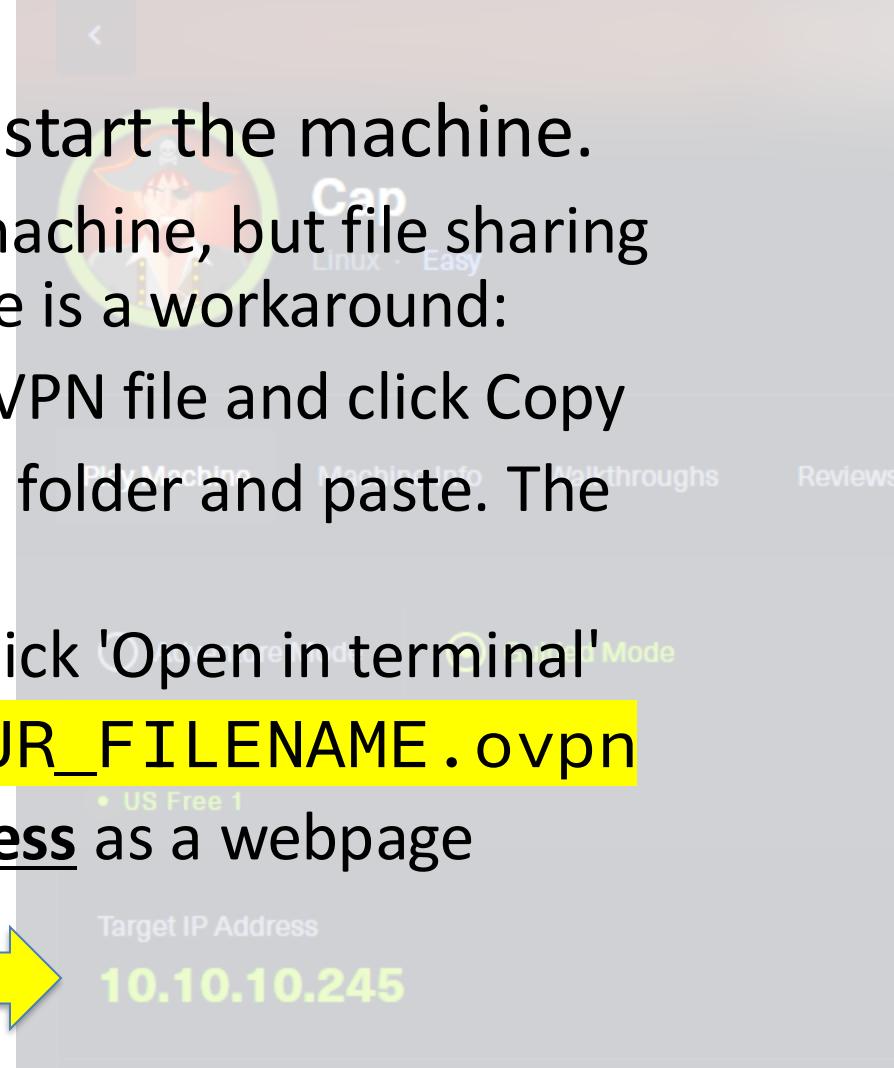
Raw hexadecimal contents of packet

0000 08 00 27 58 6e af 52 54 08 12 35 02 08 00 45 00 ... "Xn RT - 5 - E  
0010 08 28 21 7e 00 00 40 06 d5 65 b9 7d be 60 0a 00 -(!~@ - e) - P  
0020 02 0f 00 50 84 d6 04 c2 c0 bb 29 bd ad c3 50 11  
0030 ff ff 09 c2 00 00 00 00 00 00 00 00 00 00 00 00

Packets: 191 - Displayed: 191 (100.0%) - Dropped: 0 (0.0%) Profile: Default

# Cap - HTB

- [Hack The Box :: Hack The Box](#)
- Download the OpenVPN file (top right) then start the machine.
  - You need to run the OpenVPN file on your Kali machine, but file sharing between VM and your desktop is restricted. Here is a workaround:
  - On your Windows machine, right click the OpenVPN file and click Copy
  - On the Kali VM, right click on the desktop or any folder and paste. The clipboard is shared between VM and host.
  - Right click the same folder/desktop again, and click 'Open in terminal'
  - In the terminal, enter `sudo openvpn ./YOUR_FILENAME.ovpn`
  - Open Firefox in Kali and open the target IP address as a webpage



# Task Knowledge

- OSI Model – psychical, data, network, transport, session, presentation, application
- TCP/IP - link, internet, transport, application
- Linux
- Ports
- Wireshark
- tcpdump and tshark