TUTORIAL - 4

Operating Systems and Networks

Today's Tut??

Call Your Friends!

Name: _____ Surname: _____

Number: _____ Class: 6th____

Date: _____

Mark: _____

Teacher: _____

E.E: _____

Reading comprehension Test

1 Read the text carefully.

My name is Jessica and I am an Australian student. I am thirteen years old and I live in Sidney with my parents and my sister, Ruth. My mother is a nurse and my father is a doctor. They both work in a hospital.

I am a very active girl. I like playing tennis and swimming. I play tennis twice a week.

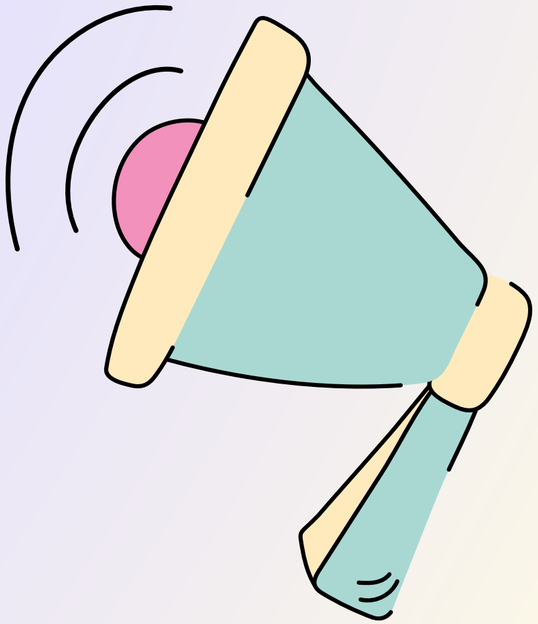
I usually get up at half past seven. I have a quick shower, get dressed and then I have breakfast with my family. After breakfast I always brush my teeth.

At twenty past eight my sister Ruth and I go to school by bus. Lessons start at nine o'clock. We have lunch in the school canteen and in the afternoon we have lessons again. We always go home at four o'clock. At five o'clock we have a snack and then we do the homework. We usually have dinner at eight o'clock and, after dinner, I watch TV or read a book. I love reading. Ruth doesn't like reading so she plays computer games. We go to bed at half past ten because we get up early.

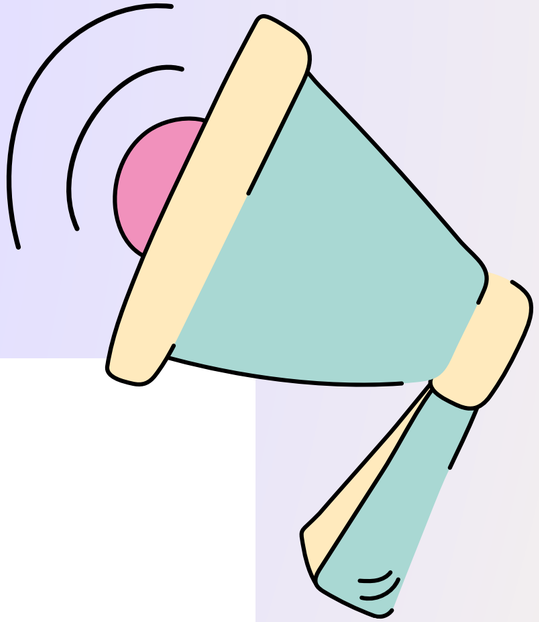
2 All these sentences are false. Correct them.

MP-2 Announcements

- **Deadline Policy:**
  - No doubts will be answered in the **last week before the deadline**.
  - You are required to accept the **GitHub Classroom invitation** immediately. Any request to accept the repository after **10 days** will not be considered.
- **Doubt Policy:**
  - Read through the requirements thoroughly **before asking doubts**.
  - No repetitive questions will be entertained.
  - You are expected to **Google or use GPT** for basic concepts.
  - **No project-related queries** will be answered outside HackMD.
- **Code Testing:**
  - Make sure to **thoroughly test your code** before submission to avoid runtime errors or unexpected behavior.
- **Late Submission Policy:**
  - Attempts to bypass GitHub Classroom deadlines (using hacks, backdating commits, etc.) will result in a **straight zero**.



MP-2 Announcements

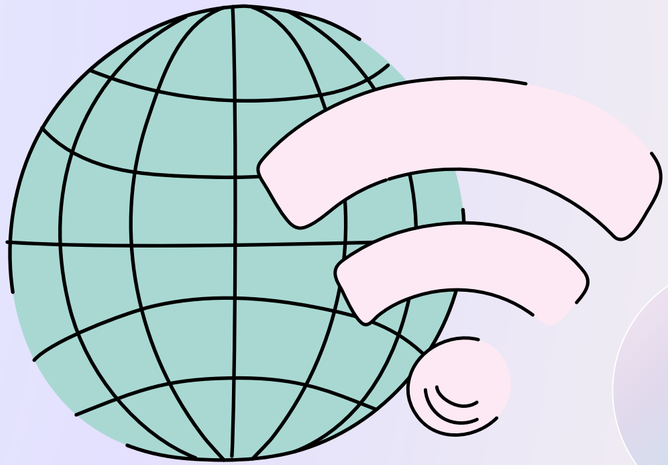


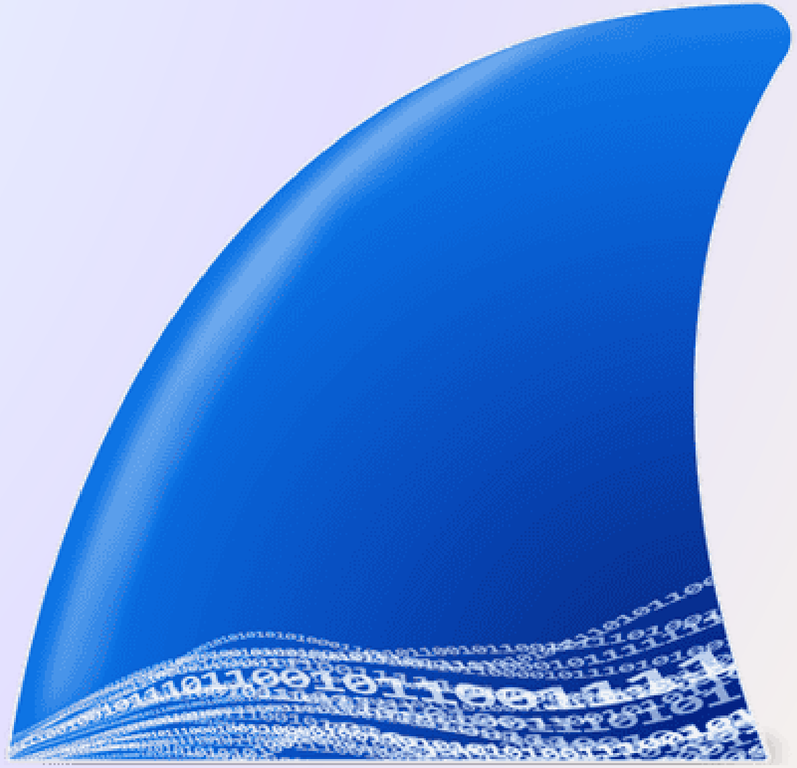
3. Notes Regarding Grading

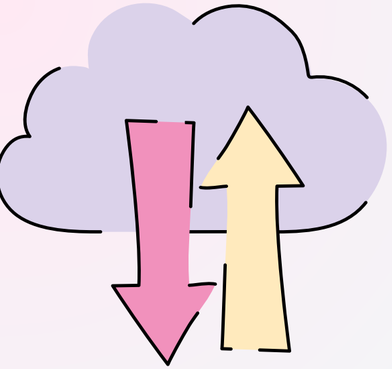
- The assignment has **three parts** with below given weights to overall course total:
  - Part A — 50%
  - Part B — 30%
  - Part C — 20%
- **Evaluation Process:**
  - In-person evaluation will hold **more weightage** than code/automated evaluation.
  - Automated evaluation will strictly check for **correct input/output formats**.
  - Any deviation from specified formats will likely result in **zero score**.

Today's Tut??

Seriously this time...

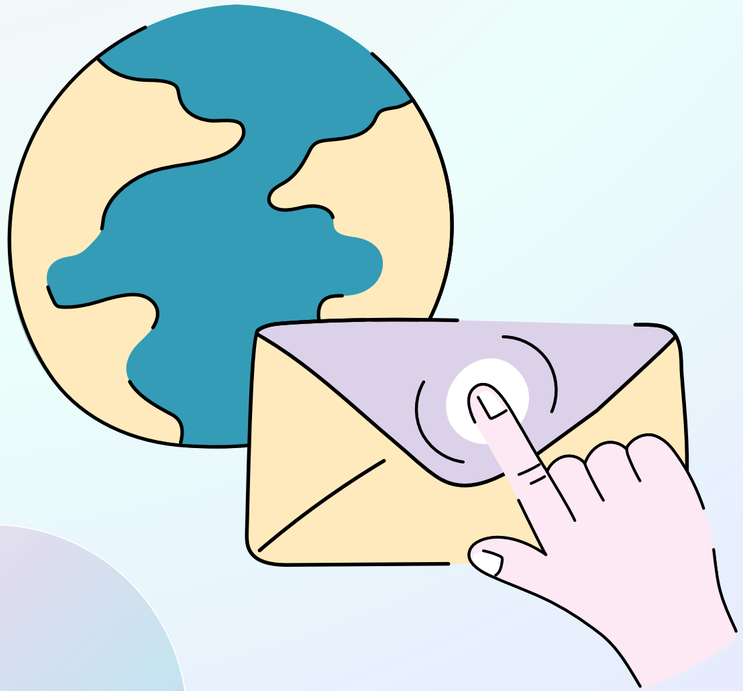






What is WireShark?

A Packet Sniffer!



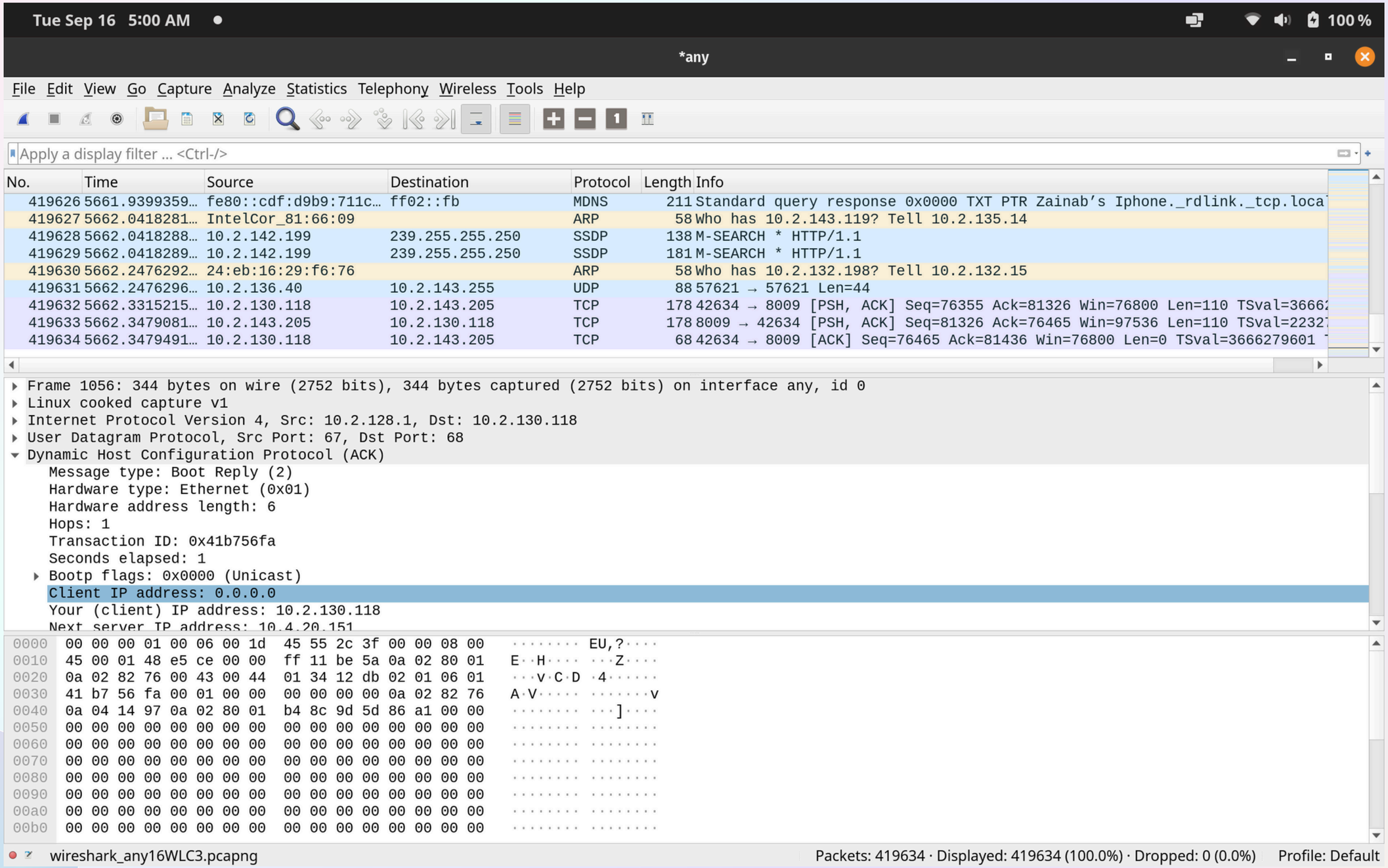
What are Packets?

Packets are small chunks of data that travel across the internet or a network. Think of them like tiny envelopes carrying information (such as a message, image, or webpage) between devices, like your computer and a server.

Each packet contains essential information like the sender’s and receiver’s address, the type of data it carries, and other details needed to successfully deliver the message.

What is Wireshark?

Wireshark is a tool used to capture and inspect these packets as they travel through a network. It acts like a "network microscope" that lets you see what’s happening inside each data packet.


Tue Sep 16 5:00 AM
100 %
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
419626 5661.9399359… fe80::cdf:d9b9:711c… ff02::fb MDNS 211 Standard query response 0x0000 TXT PTR Zainab’s Iphone._rdlink._tcp.loca
419627 5662.0418281… IntelCor_81:66:09 ARP 58 Who has 10.2.143.119? Tell 10.2.135.14
419628 5662.0418288… 10.2.142.199 239.255.255.250 SSDP 138 M-SEARCH * HTTP/1.1
419629 5662.0418289… 10.2.142.199 239.255.255.250 SSDP 181 M-SEARCH * HTTP/1.1
419630 5662.2476292… 24:eb:16:29:f6:76 ARP 58 Who has 10.2.132.198? Tell 10.2.132.15
419631 5662.2476296… 10.2.136.40 10.2.143.255 UDP 88 57621 → 57621 Len=44
419632 5662.3315215… 10.2.130.118 10.2.143.205 TCP 178 42634 → 8009 [PSH, ACK] Seq=76355 Ack=81326 Win=76800 Len=110 TSval=3666
419633 5662.3479081… 10.2.143.205 10.2.130.118 TCP 178 8009 → 42634 [PSH, ACK] Seq=81326 Ack=76465 Win=97536 Len=110 TSval=2232
419634 5662.3479491… 10.2.130.118 10.2.143.205 TCP 68 42634 → 8009 [ACK] Seq=76465 Ack=81436 Win=76800 Len=0 TSval=3666279601
Frame 1056: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.2.128.1, Dst: 10.2.130.118
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (ACK)
Message type: Boot Reply (2)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 1
Transaction ID: 0x41b756fa
Seconds elapsed: 1
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 10.2.130.118
Next server IP address: 10.4.20.151
0000 00 00 00 01 00 06 00 1d 45 55 2c 3f 00 00 08 00 ···· EU,?····
0010 45 00 01 48 e5 ce 00 00 ff 11 be 5a 0a 02 80 01 E··H··· ···Z····
0020 0a 02 82 76 00 43 00 44 01 34 12 db 02 01 06 01 ···v·C·D ·4·····
0030 41 b7 56 fa 00 01 00 00 00 00 00 00 0a 02 82 76 A·V····· ·····v
0040 0a 04 14 97 0a 02 80 01 b4 8c 9d 5d 86 a1 00 00 ···· ···]····
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ···· ····
wireshark_any16WLC3.pcapng
Packets: 419634 · Displayed: 419634 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

Why bother?

Detailed Packet Capture

Captures all network packets in real-time for deep analysis.

Supports Many Protocols

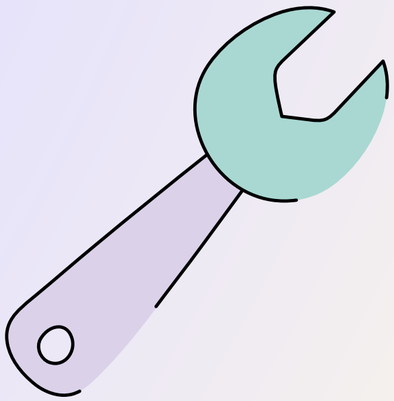
Understands and decodes hundreds of network protocols automatically.

Easy to Use with GUI

User-friendly interface to filter, inspect, and follow network traffic.

Helps Troubleshoot Network Issues

Identifies slow connections, dropped packets, and suspicious activity.



Free and Open Source

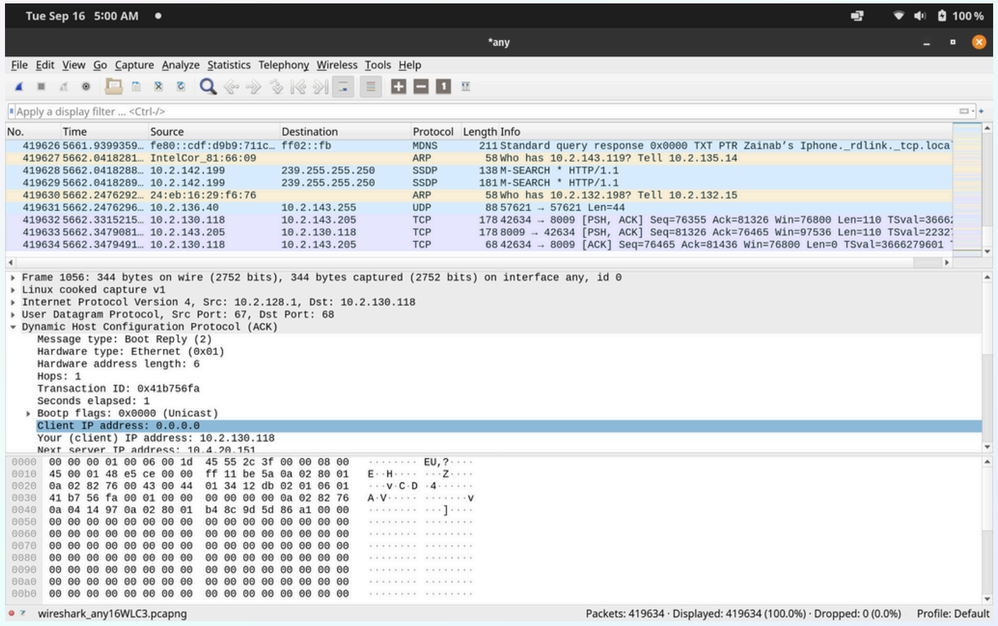
Completely free to use and supported by a large community.

and also...

Part B — The Terminal Packet Sniffer

> 🦈 **Welcome to the C-Shark Division!**
> This is **LAZY Corp's** idea of a cybersecurity team. Forget expensive firewalls or fancy monitoring tools — you'll be handed a **terminal-based shark fin** and told to sniff out suspicious packets.
> Your task is to build C-Shark, a terminal-only sniffer that LAZY Corp swears is "just as good as Wireshark" (legal says we have to stop calling it "diet Wireshark"). With it, you'll see everything flowing through the network: shady MAC addresses, questionable IP headers, and DNS queries that definitely don't look work-related.
> Think of it as giving you x-ray specs for the internet, only instead of superheroes, you're an underpaid intern staring at hex dumps.

:)




```
divijhmangtani@pop-os:~/IIIT-H/Sem5/OSNTa/mp2$ sudo ./c-shark
[sudo] password for divijhmangtani:
[C-Shark] Enhanced Network Protocol Analyzer
================================================
[C-Shark] Found devices:
1. wlan0
2. any (Pseudo-device that captures on all interfaces)
3. lo
4. docker0
5. bluetooth0 (Bluetooth adapter number 0)
6. bluetooth-monitor (Bluetooth Linux Monitor)
7. nflog (Linux netfilter log (NFLOG) interface)
8. nfqueue (Linux netfilter queue (NFQUEUE) interface)
9. dbus-system (D-Bus system bus)
10. dbus-session (D-Bus session bus)
Select an interface to sniff (1-10):
```

```
Select option (or Ctrl+D to exit): 3

[C-Shark] Filter Menu
====================
1. HTTP Traffic (port 80)
2. HTTPS Traffic (port 443)
3. TCP Traffic
4. UDP Traffic
5. ICMP Traffic
6. ARP Traffic
7. SSH Traffic (port 22)
8. DNS Traffic (port 53)
9. Back to Main Menu

Select filter:
```

```
Select option (or Ctrl+D to exit): Invalid option. Please try aga

[C-Shark] Network Protocol Analyzer - Interface: wlan0
=====================================================
Main Menu:
1. Start Monitoring (All Packets)
2. Start Monitoring with Hex Display
3. Start Monitoring with Filters
4. Inspect Stored Packets
5. Exit

Select option (or Ctrl+D to exit):
```

```
----------------------------------------
Packet #86 | Timestamp: 1757344403.
  L2 (Ethernet): Dst MAC: A2:E2:62:
  L3 (IPv6):     Src IP: 2409:40f0:
                 Traffic Class: 0 |
  L4 (TCP):      Src Port: 35230 |
                 Window: 568 | Checksum: 0x0791 | Header Length: 32 bytes
  L7 (Payload):  Identified as HTTPS/TLS on port 443 - 35 bytes
    Data (first 35 bytes):
    17 03 03 00 1E 4A 7E 8B 3F A3 72 F0 E5 4F CF C4   .....J~.?.r..O..
    63 54 B2 2B F7 05 C2 82 B9 A9 78 F4 7E 85 B2 90   cT.+......x.~...
    09 3F 97                                          .?.
    [TLS/SSL encrypted data]
----------------------------------------
Packet #87 | Timestamp: 1757344403.422121 | Length: 125 bytes
  L2 (Ethernet): Dst MAC: A2:E2:62:90:AE:15 | Src MAC: B4:8C:9D:5D:86:A1 | EtherType: IPv6 (0x86DD)
  L3 (IPv6):     Src IP: 2409:40f0:12e:b820:d105:b3d0:30f:b9b6 | Dst IP: 2404:6800:4002:802::200e | Next Header: TCP (6) | Hop Limit: 64
                 Traffic Class: 0 | Flow Label: 0x00000 | Payload Length: 71
  L4 (TCP):      Src Port: 35230 | Dst Port: 443 (HTTPS) | Seq: 3969662159 | Ack: 1383627305 | Flags: [ACK,PSH]
                 Window: 568 | Checksum: 0x2160 | Header Length: 32 bytes
  L7 (Payload):  Identified as HTTPS/TLS on port 443 - 39 bytes
    Data (first 39 bytes):
    17 03 03 00 22 C1 06 F1 BB 4C 8F D6 95 34 6E 09   ....".....L...4n.
    93 6F 3C AE C3 9C 8D A5 AE DA 93 EA DD 50 E5 2D   .o<..........P.-
    1E A9 21 65 BC BE 22                              ..!e..."
    [TLS/SSL encrypted data]
----------------------------------------
Packet #88 | Timestamp: 1757344403.496965 | Length: 86 bytes
  L2 (Ethernet): Dst MAC: B4:8C:9D:5D:86:A1 | Src MAC: A2:E2:62:90:AE:15 | EtherType: IPv6 (0x86DD)
  L3 (IPv6):     Src IP: 2404:6800:4002:802::200e | Dst IP: 2409:40f0:12e:b820:d105:b3d0:30f:b9b6 | Next Header: TCP (6) | Hop Limit: 120
                 Traffic Class: 0 | Flow Label: 0x00000 | Payload Length: 32
  L4 (TCP):      Src Port: 443 (HTTPS) | Dst Port: 35230 | Seq: 1383627305 | Ack: 3969662159 | Flags: [ACK]
                 Window: 987 | Checksum: 0xF723 | Header Length: 32 bytes
----------------------------------------
Packet #89 | Timestamp: 1757344403.496966 | Length: 86 bytes
  L2 (Ethernet): Dst MAC: B4:8C:9D:5D:86:A1 | Src MAC: A2:E2:62:90:AE:15 | EtherType: IPv6 (0x86DD)
  L3 (IPv6):     Src IP: 2404:6800:4002:802::200e | Dst IP: 2409:40f0:12e:b820:d105:b3d0:30f:b9b6 | Next Header: TCP (6) | Hop Limit: 120
                 Traffic Class: 0 | Flow Label: 0x00000 | Payload Length: 32
  L4 (TCP):      Src Port: 443 (HTTPS) | Dst Port: 35230 | Seq: 1383627305 | Ack: 3969662198 | Flags: [ACK]
                 Window: 987 | Checksum: 0xF6FC | Header Length: 32 bytes
----------------------------------------
^C
[C-Shark] Monitoring stopped. Returning to menu...
[C-Shark] Monitoring stopped. Captured 89 packets.

[C-Shark] Network Protocol Analyzer - Interface: wlan0
=====================================================
Main Menu:
1. Start Monitoring (All Packets)
2. Start Monitoring with Hex Display
3. Start Monitoring with Filters
4. Inspect Stored Packets (Last session: 89 packets)
5. Exit

Select option (or Ctrl+D to exit):
```

```
.mit: 64

  Header Checksum: 0x5E8D (Bytes 24-25)
    └─ Hex: 5E 8D
  Source IP:       127.0.0.1 (Bytes 26-29)
    └─ Hex: 7F 00 00 01
  Destination IP:  127.0.0.1 (Bytes 30-33)
    └─ Hex: 7F 00 00 01

🚚 TCP HEADER (Layer 4)
──────────────────────
  Source Port:     8080 (Bytes 34-35)
    └─ Hex: 1F 90
  Destination Port: 59314 (Bytes 36-37)
    └─ Hex: E7 B2
  Sequence Number: 3204926769 (Bytes 38-41)
    └─ Hex: BF 07 4D 31
  Acknowledgment:  3160439224 (Bytes 42-45)
    └─ Hex: BC 60 79 B8
  Header Length:   32 bytes (8 * 4) (Upper 4 bits of byte 46)
    └─ Hex: 80 (upper 4 bits = 8)
  Flags:           0x18 (Byte 47)
    └─ Hex: 02 00
    └─ URG:0 ACK:1 PSH:1 RST:0 SYN:0 FIN:0
  Window Size:     512 (Bytes 48-49)
    └─ Hex: 02 00
  Checksum:        0x1150 (Bytes 50-51)
    └─ Hex: 11 50
  Urgent Pointer:  0 (Bytes 52-53)
    └─ Hex: 00 00
  TCP Options:     12 bytes (Bytes 54-65)
    └─ Hex: 01 01 08 0A 20 DC 18 50 20 DC 18 4F

📄 APPLICATION DATA (Layer 5-7)
──────────────────────────────
  Payload Length:  4903 bytes (Bytes 66-4968)
  Protocol:        Unknown/Custom (Port 8080)

  First 64 bytes of payload:
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    ASCII

  0042 23 20 43 6F 6D 70 75 74 65 72 20 47 72 61 70 68   # Computer Graph
  0052 69 63 73 20 41 73 73 69 67 6E 6D 65 6E 74 20 23   ics Assignment #
  0062 33 0A 0A 23 23 20 4F 76 65 72 76 69 65 77 0A 0A   3..## Overview..
  0072 54 68 69 73 20 61 73 73 69 67 6E 6D 65 6E 74 20   This assignment

  ... and 4839 more bytes

  ═══════════════ END OF PACKET ANALYSIS ═══════════════

Press Enter to continue...
```

Reminder

Creative Freedom: The example input/output formats shown below are just that—examples! You have the freedom to design your own interface, as long as all the required functionality is present and the output is clear and readable.

Reconnaissance is Key: It is highly recommended to read through all the project phases before writing a single line of code. A good plan will help you structure your code in a modular and expandable way from the very beginning.

Choosing Your Hunting Ground: College Wi-Fi or corporate LANs can have complex configurations (it will and should still work on it, but the packets on there may be less predictable). For easier debugging, it's a great idea to use your own personal hotspot. The packets will be much more predictable! Even if you don't have hotspot connection, you can set up and try to debug using localhost (`lo` interface - a localserver) for testing purposes.

Root Privileges Required: Packet sniffing requires deep access to the network stack. You will need to run your final executable with sudo for it to work.

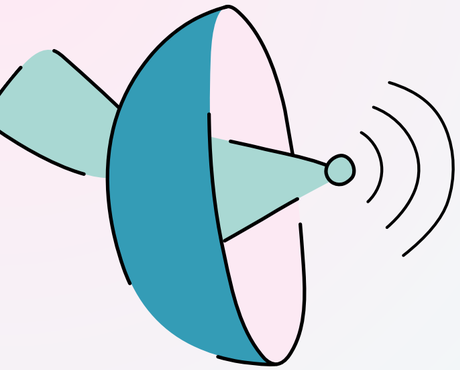
```
sudo ./cshark
```

A Shark, Not a Kraken: Remember your role. You are a silent predator, observing from the depths. Your job is to watch the traffic flow by, not to thrash about and create a tidal wave. C-Shark is a *listener*, not a talker. You are strictly forbidden from sending, injecting, or crafting packets. LAZY Corp's legal team has a very small budget, and they don't want to spend it bailing you out for taking down the campus Wi-Fi. Observe only.

Download

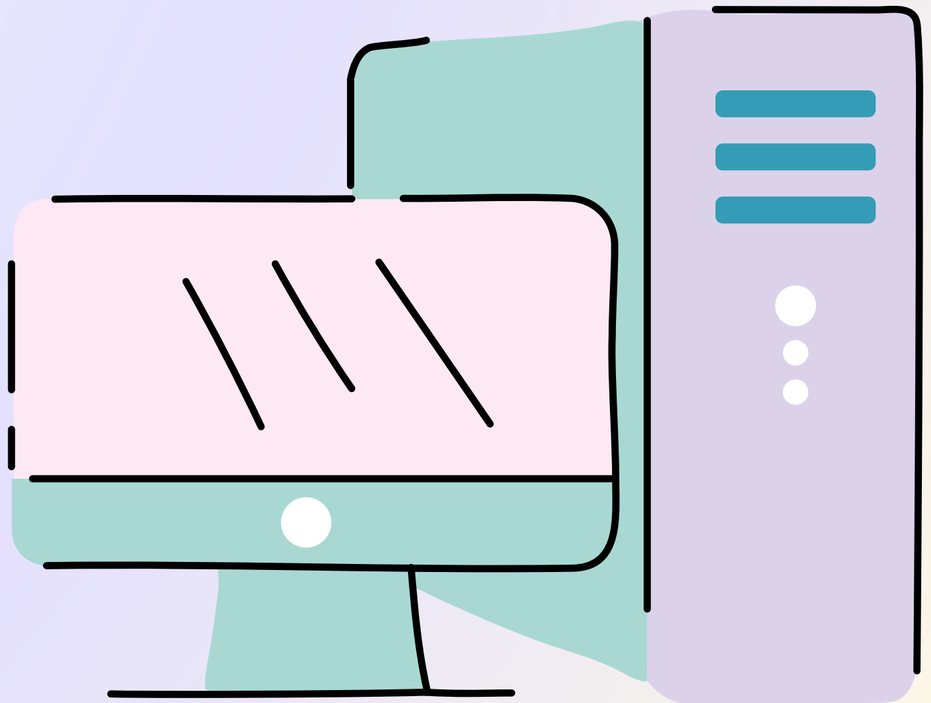
https://www.wireshark.org/download.html

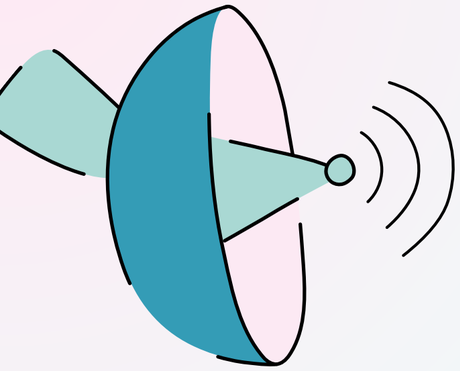
Linux - sudo apt install wireshark



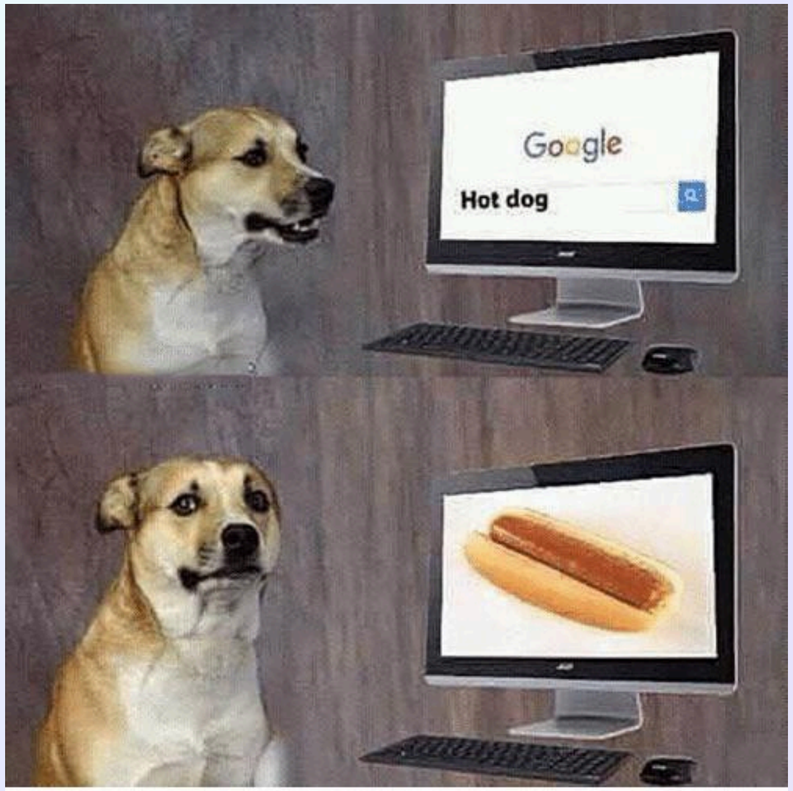
Sniffing

People could be listening

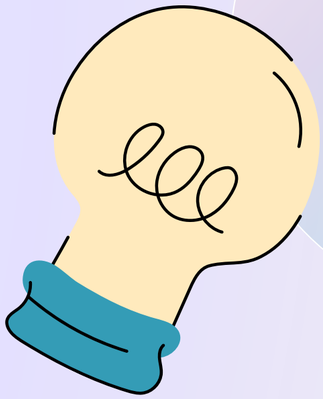




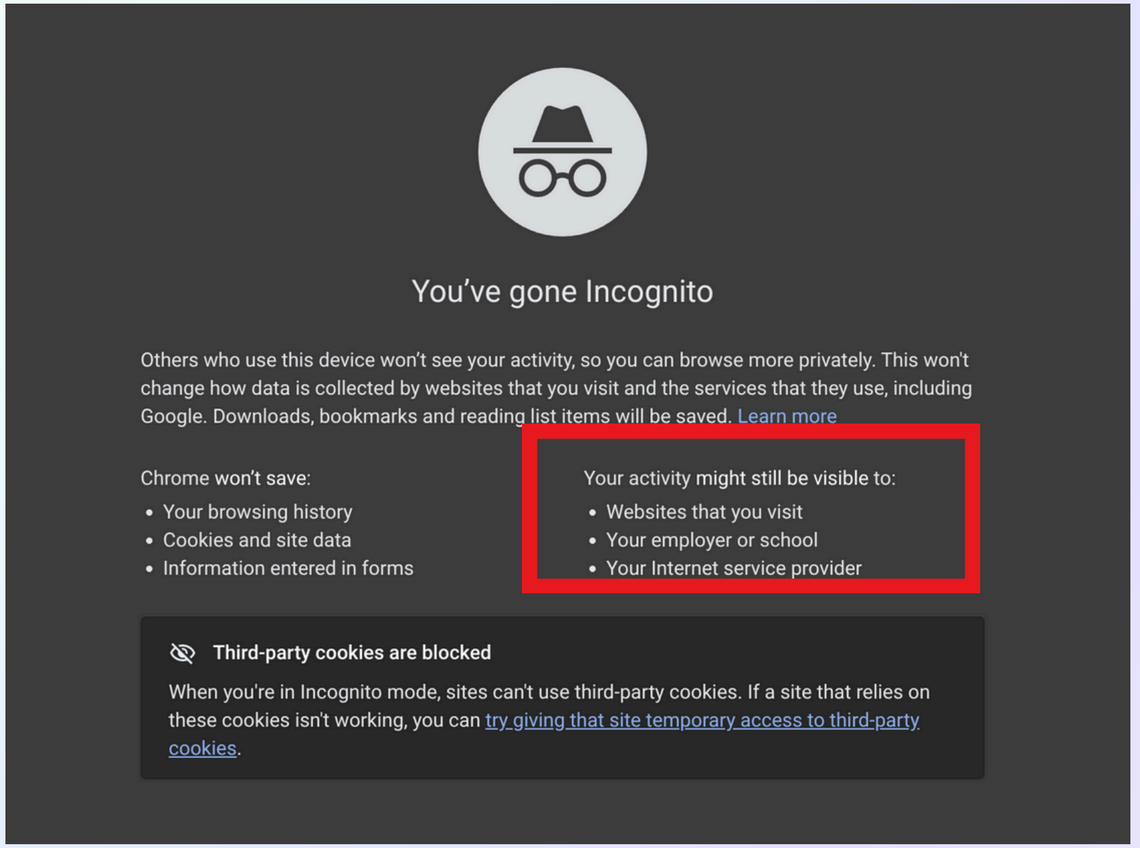
Does that mean People can see everything?



But I use Incognito!

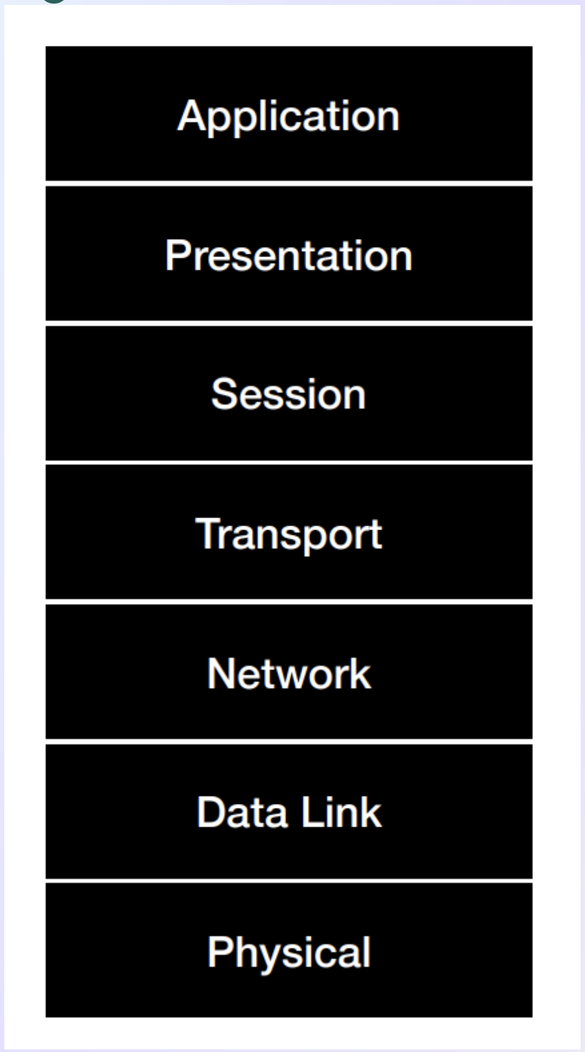


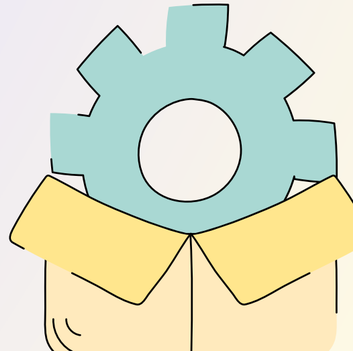
We'll be seeing whats directly and in-directly visible by packet sniffing today!



Quick Recap First

The holy grail OSI



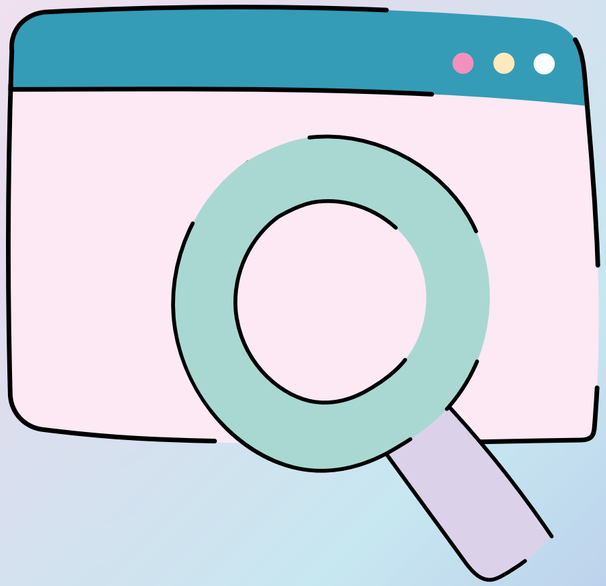


Lets Inspect in Wireshark👀

Layer by Layer as in MP2 (and in terminal)

- L2
- L3- Ipv4, Ipv6, ARP
- L4- TCP, UDP...ports?
- Payload - HTTP, HTTPS, DNS

Examples for all



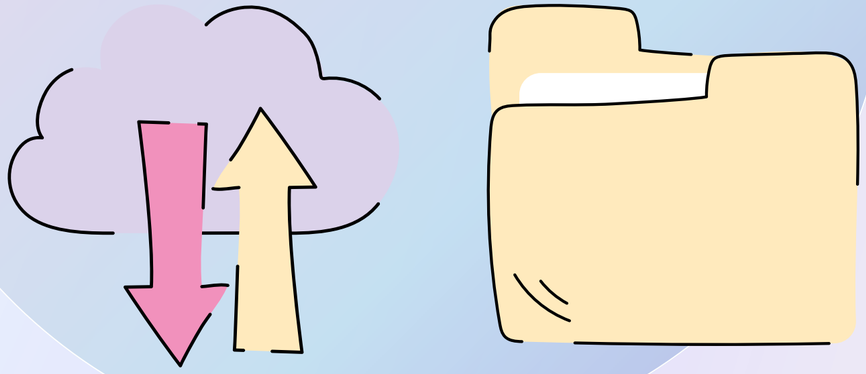
So is the payload always visible?

HTTP VS HTTPS

Example

So what is still visible?

Does HTTPS encrypt EVERYTHING?



IP addresses

- Filter and track example
- DNS and reverse DNS using dig

So what you visit is visible, but not what you did there

Time-to-Live (TTL) / Hop Limit

What is TTL / Hop Limit?

- TTL in IPv4 and Hop Limit in IPv6 control the lifespan of a packet on a network.
- These fields prevent routing loops by limiting the number of hops (routers) a packet can take before being discarded.

How TTL / Hop Limit Works:

1. Initial Value: Packet starts with a TTL/Hop Limit (e.g., 64).
2. Decrements at Each Router: Each router the packet passes through decreases the TTL by 1.
3. When TTL/Hop Limit Reaches 0: The packet is discarded, and the source gets notified.

TTL in IPv4:

- Field: 1 byte (8 bits).
- Function: Limits the number of hops a packet can take in an IPv4 network.
- Default Values: 64, 128, 255.

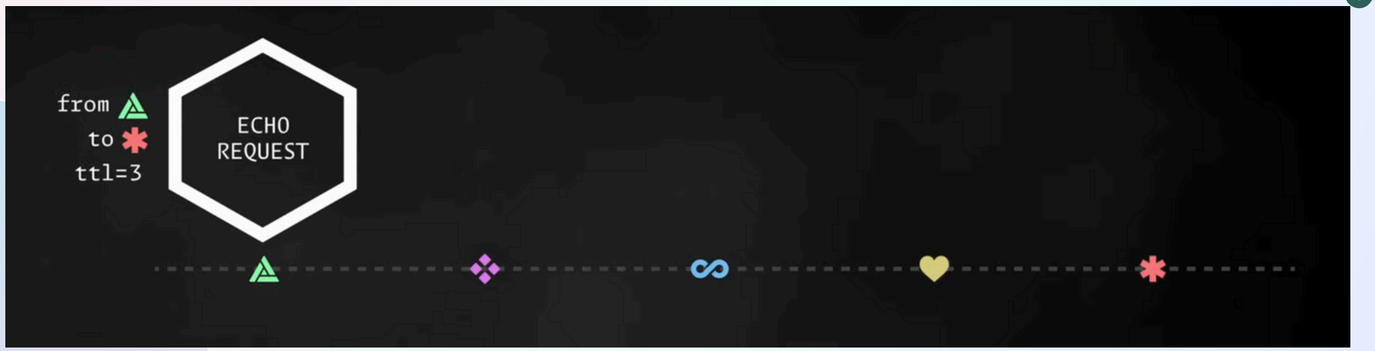
Hop Limit in IPv6:

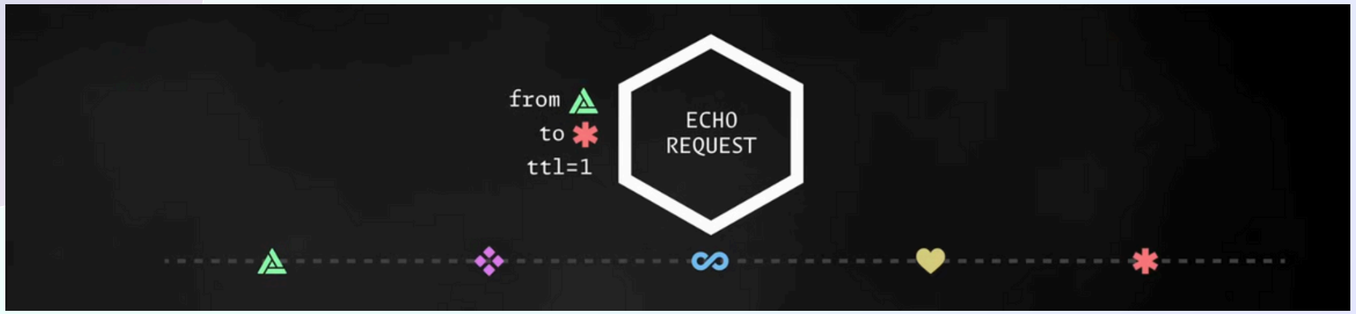
- Field: 1 byte (8 bits).
- Function: Same as TTL but more explicit (limits hops in IPv6).
- Default Values: 64, 128.

Why is TTL / Hop Limit Important?

- Prevents routing loops: Without TTL/Hop Limit, packets could endlessly loop in a network.
- Improves network performance: Ensures packets don't waste resources on endless hops.

Visually


from
to
ttl=3
ECHO
REQUEST


from
to
ttl=1
ECHO
REQUEST


from
to
ttl=9
TIME
EXCEEDED!


from
to
ttl=5
TIME
EXCEEDED!
oh
no

It can also be used to

How Traceroute Uses TTL to Track Network Path

- TTL Increment: Traceroute starts with a packet having a TTL of 1 and increases it by 1 for each successive packet.
- Each Router's Role: When a router receives a packet, it reduces the TTL by 1. When TTL = 0, the router discards the packet and sends an **ICMP "Time Exceeded" message** back.
- Hop-by-Hop Process: Traceroute gathers the information from each hop (router) by observing the TTL values.

Step-by-Step Process of Traceroute:

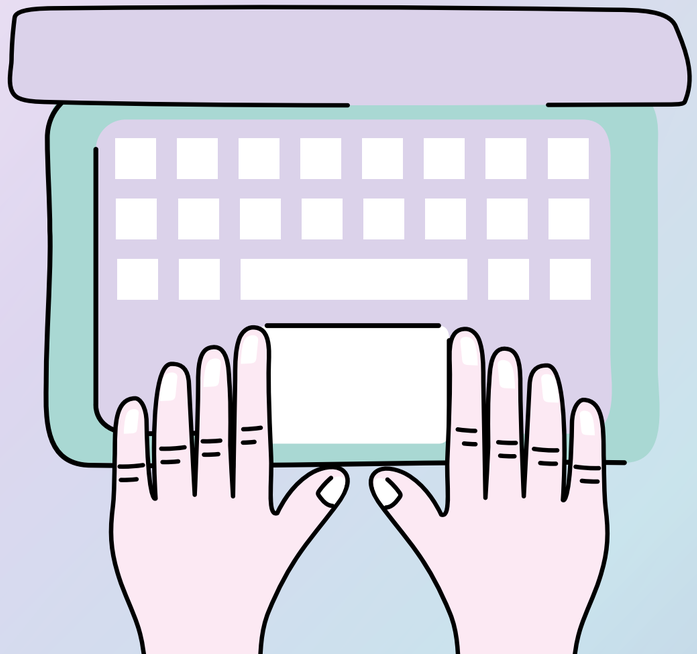
- Step 1: Send packet with TTL = 1 to the destination.
  - First router decreases TTL to 0, discards the packet, and replies with ICMP Time Exceeded.
- Step 2: Increase TTL to 2, send the packet again.
  - Second router reduces TTL to 1, packet reaches the third router, and so on.
- Step 3: Continue this process until the destination is reached.

Why TTL is Important in Traceroute

- Path Mapping: Identifies each hop the packet takes from source to destination.
- Network Troubleshooting: Helps diagnose slow or broken network paths by showing delays and route issues.
- Route Visualization: TTL increments make it easy to visualize each step and pinpoint potential network problems.

Real-World Usage of Traceroute

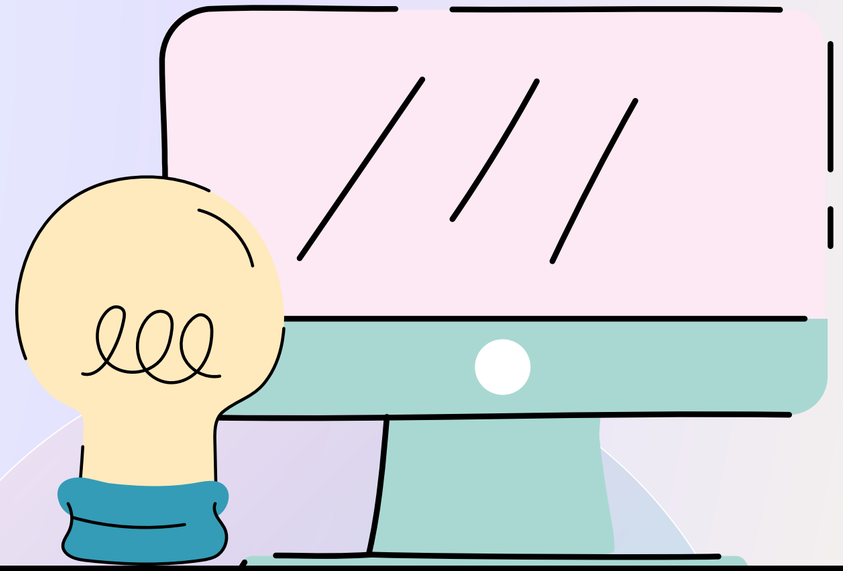
- Troubleshooting: Diagnosing where delays or failures occur in a network.
- Network Performance Monitoring: Analyzing network performance and identifying congestion points.
- Routing Issues: Locating routing loops or misconfigured routers.



Visually

https://youtu.be/jjKFXlFNR4E?si=5JhZ26xMd_TXTNxT&t=384

A video if interested



https://www.youtube.com/watch?v=jjKFXlFNR4E

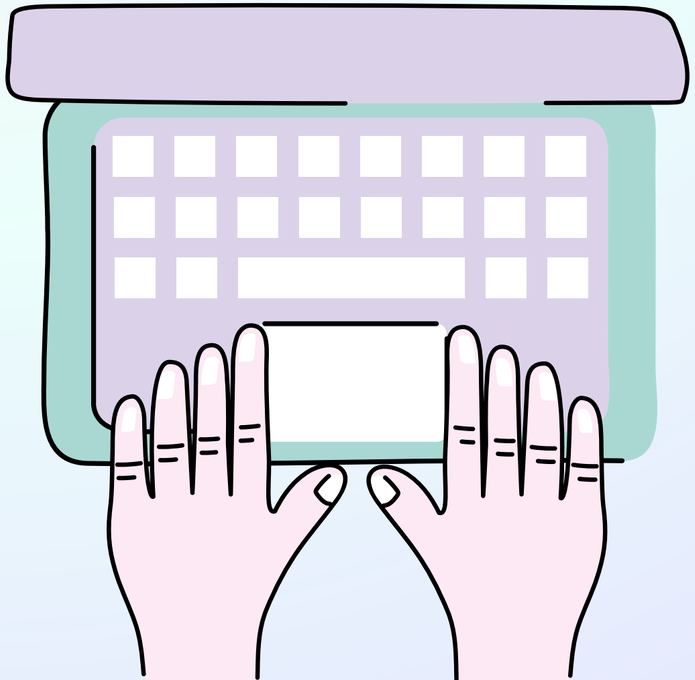
POV: I'm on my third coffee and you asked me how the internet works

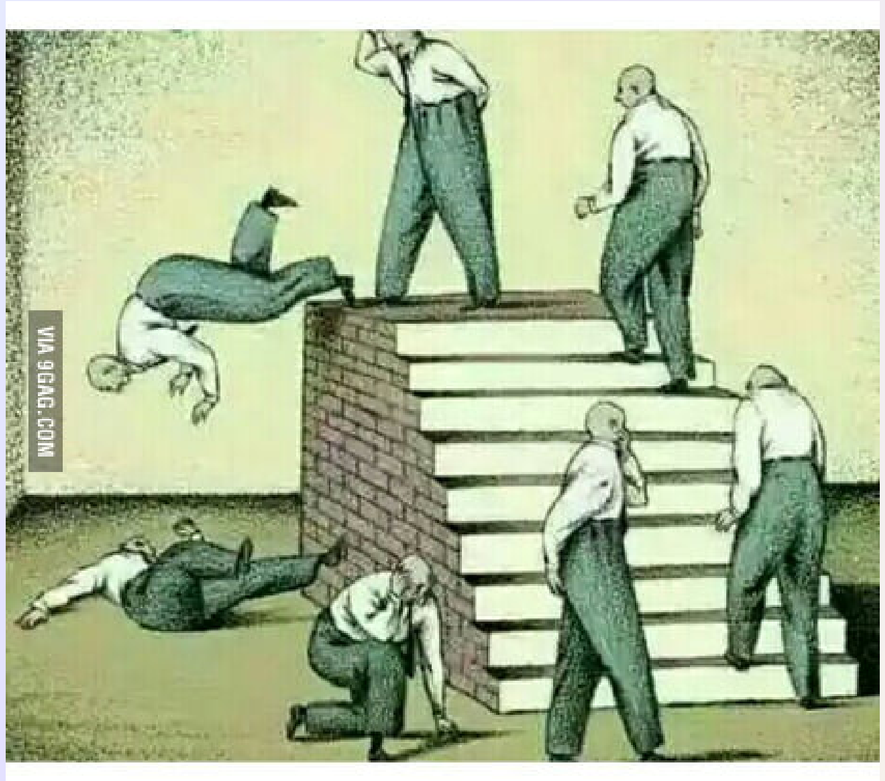
- Faster than lime

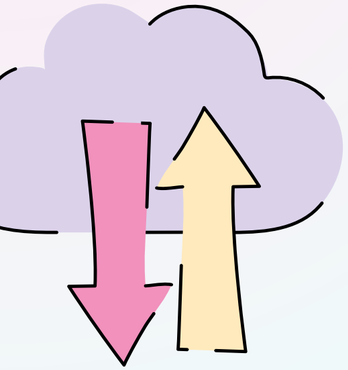
Not a rick roll XD

Don't Repeat Mistakes

Start on Time!







Thank You!