

# Week 11

# Networking

Minh Duong



# Announcements

Shib auth, we are in need of maintainer/s

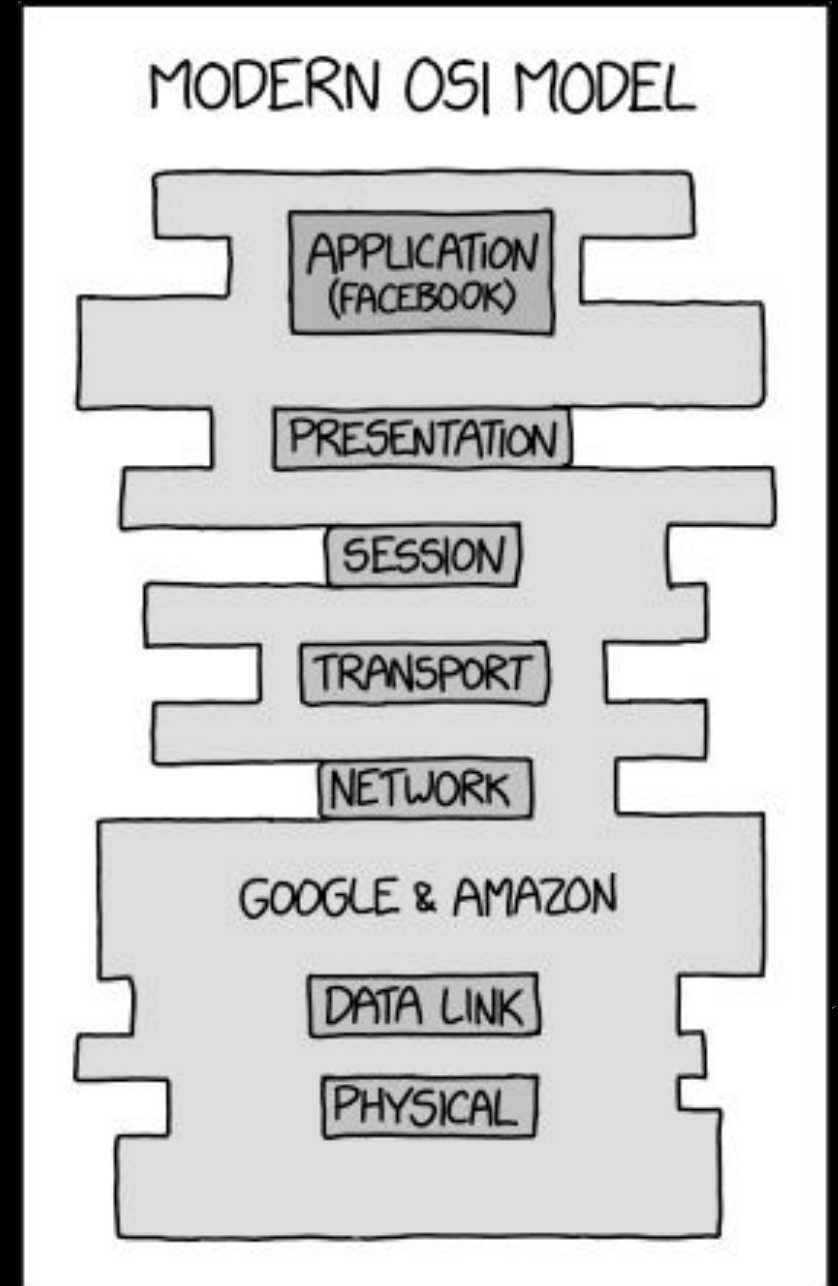
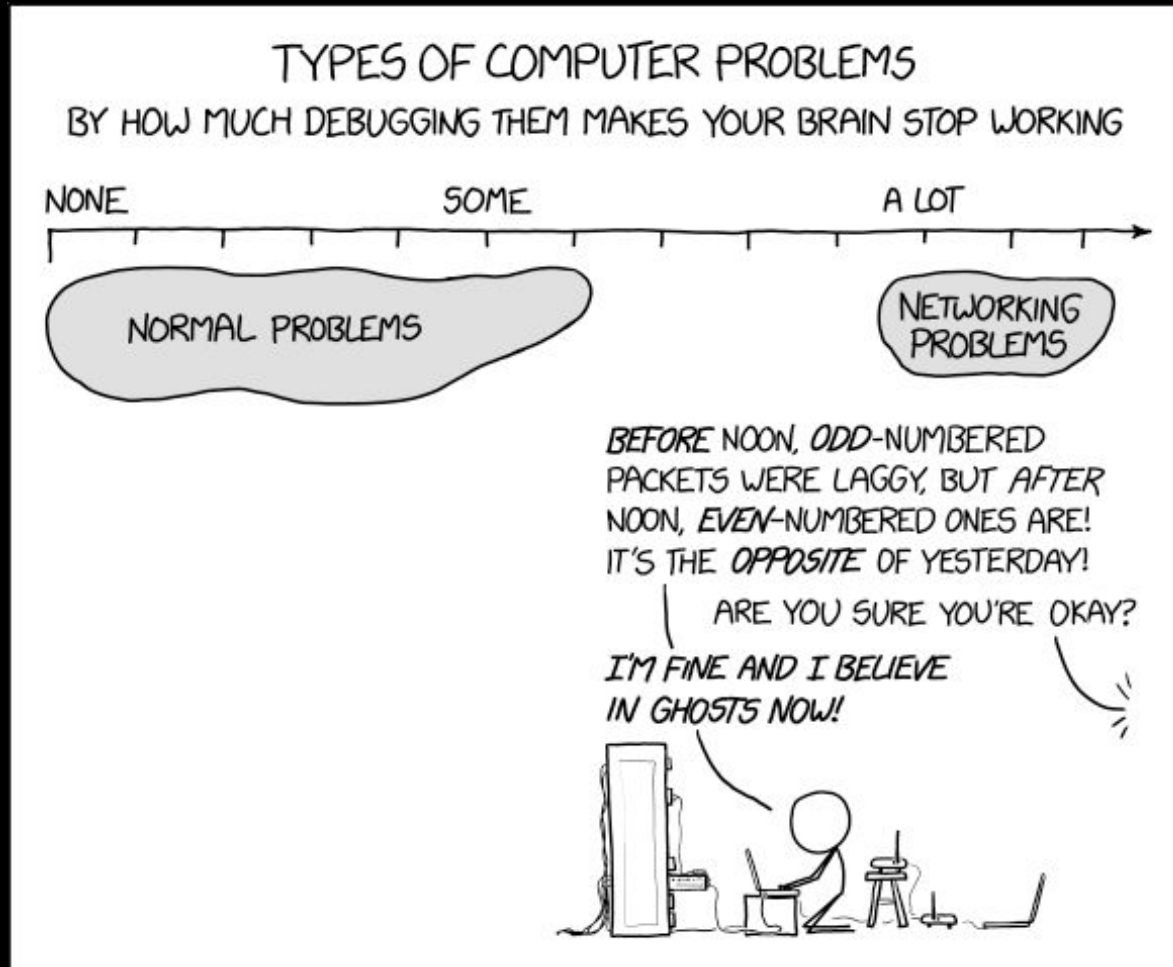
Website: we also need maintainers

Merch form now: [sigpwny.com/merch](https://sigpwny.com/merch)

Spray paint social @ some point

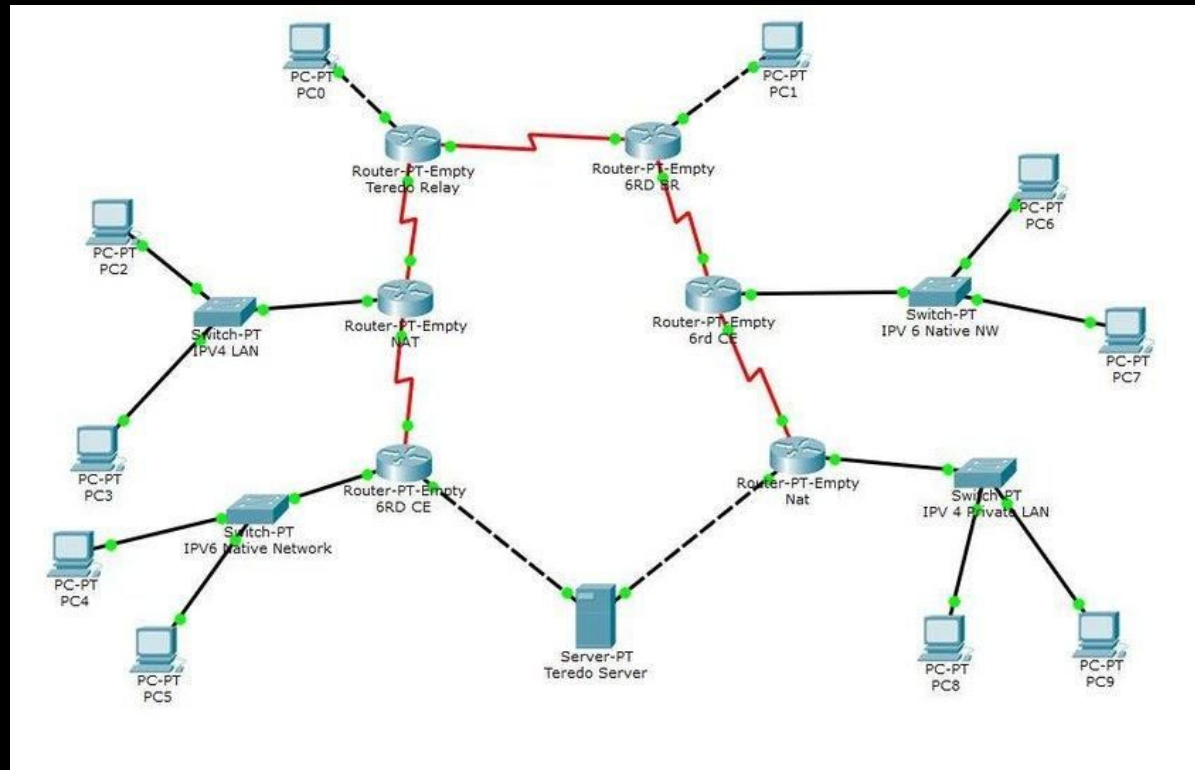


```
sigpwny{please_do_
not_throw_sausage_
pizza_away}
```



# What is Networking?

- A way for computers to send information to each other
- The Internet is only one example of a network
- Networks can have subnetworks



# Protocols for Everything

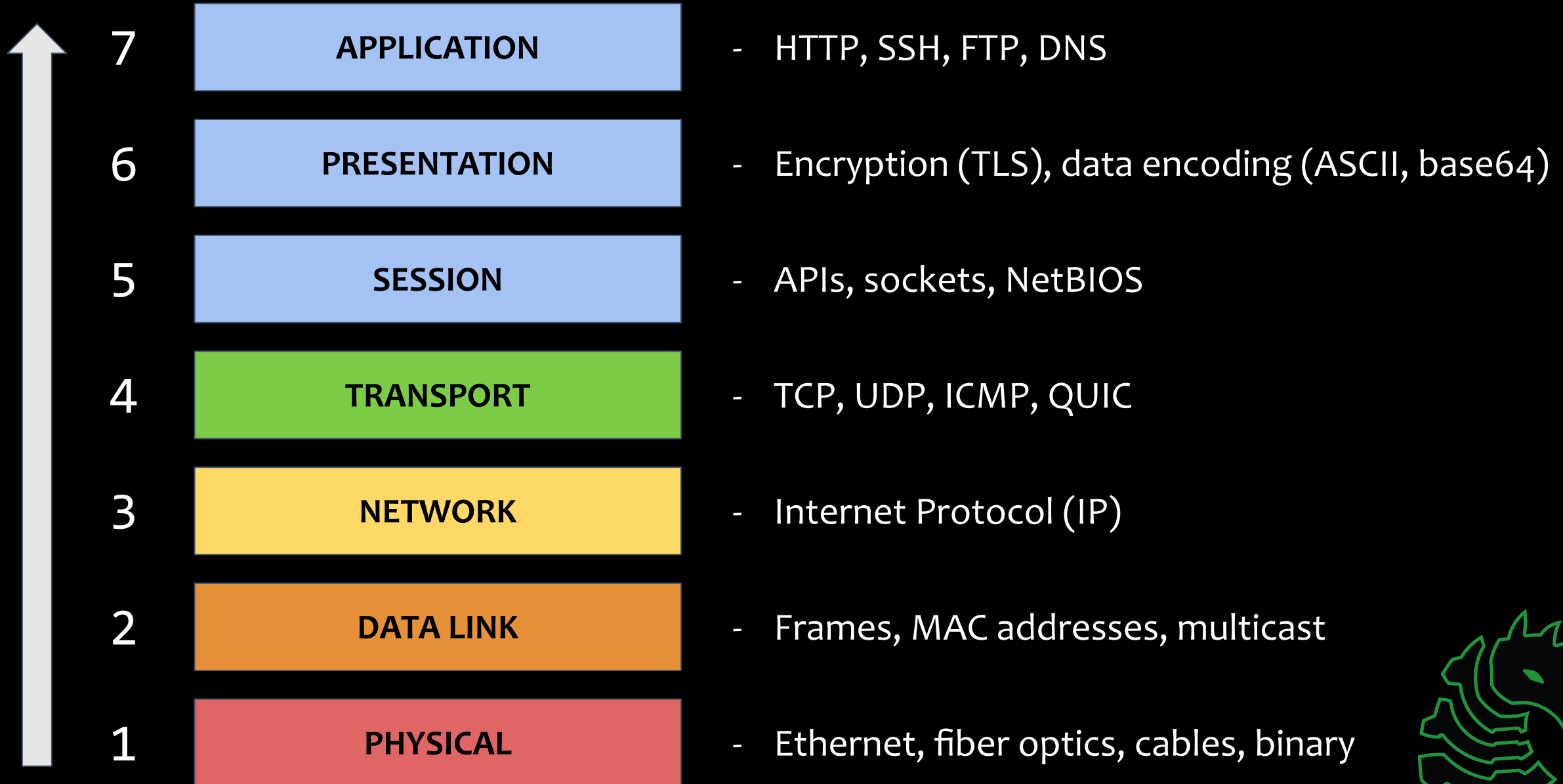
- If devices all speak different networking languages, then they can't understand each other
- As a result, protocols and standards are needed
- There are lots of networking protocols... and a lot of acronyms



# The OSI Model

- Stands for "Open Systems Interconnection"
- Breaks aspects of networking into 7 different layers
- Each layer is abstract from the other (e.g. layer 7 does not have to worry how layers 1-6 work)





# TCP vs. UDP

Imagine you want to call someone:

- TCP would be a normal conversation
  - A->B: "Hello, it's A"
  - B->A: "Oh, hi, it's B"
  - A->B: "I want to tell you something..."
- UDP would be a voicemail
  - A->B: "We've been trying to reach you about your car's warranty..."
  - No guarantee that data is received





# TCP vs. UDP

- TCP uses a three-way handshake
  - A->B: SYN
  - B->A: SYN-ACK
  - A->B: ACK
- TCP ensures reliable delivery of data
- More secure since established connection is required
- UDP just constantly streams the data
  - Useful for low-latency games or video streaming
  - There is no guarantee that you will receive the data



# Network Attacks



# SYN Flood

- Attack abusing TCP functionality
  - Attacker sends "SYN" and server responds with "SYN-ACK"
  - Server waits for "ACK" but it never comes and after a while it times out
- 
- If an attacker sends a lot of SYN packets, server will keep responding and waiting for ACK until it is handling too many connections
  - Eventually starts dropping connections and legitimate traffic cannot connect



# Arp Cache Poisoning

Who is 1.2.3.4???

Hello I am 1.2.3.4, my mac address is AA:BB:CC:DD:EE:FF

Hello I am 1.2.3.4, my mac address is 00:11:22:33:44:55

Ok I will save 1.2.3.4 as AA:BB:CC:DD:EE:FF

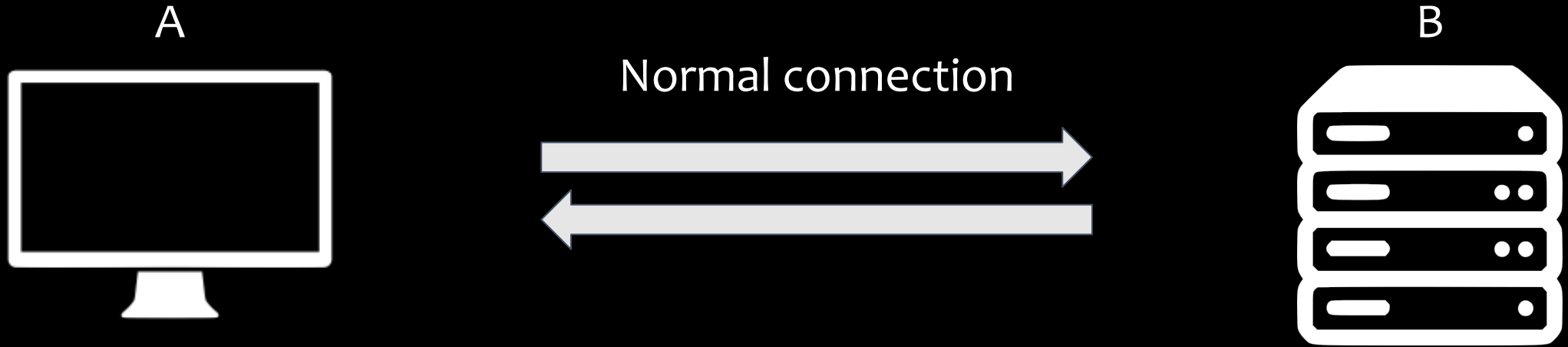


# Man-in-the-Middle (MITM)

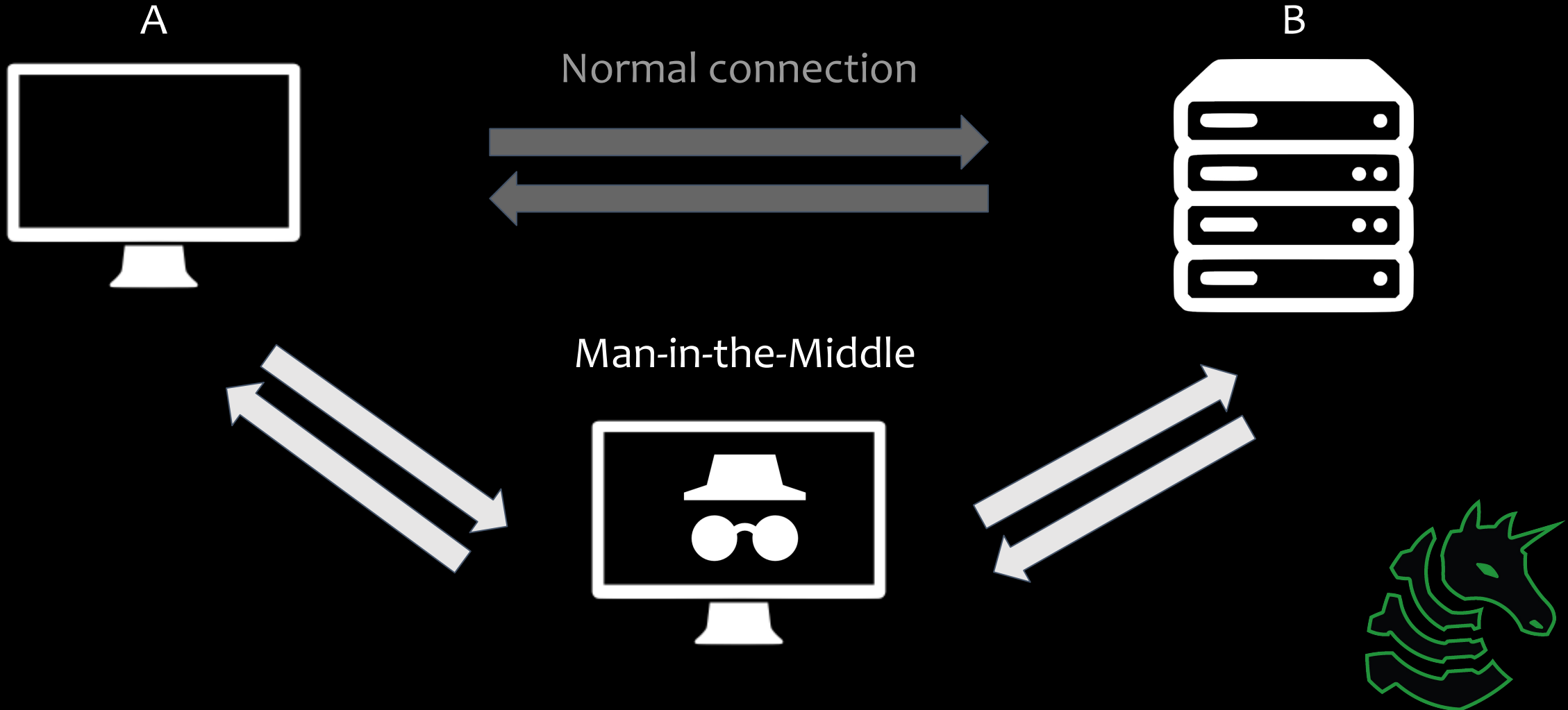
- An entity that intercepts network traffic between two parties, usually without them knowing
- Two types:
  - Passive - read data only
  - Active - modify data and resend it
- Your ISP can be considered as a MITM



# Man-in-the-Middle (MITM)



# Man-in-the-Middle (MITM)



7

APPLICATION

- Basically web/pwn

6

PRESENTATION

- Basically crypto

5

SESSION

- Session sniffing

4

TRANSPORT

- DDoS, SYN flood

3

NETWORK

- DDoS, ARP poisoning

2

DATA LINK

- MAC address spoofing

1

PHYSICAL

- Destroying physical cables

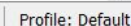




# Wireshark

- Captures all packets being sent and saves them
- Analyze packets for information
- Use cases:
  - Finding information a packet contains (e.g. plaintext credentials sent over HTTP)
  - Network forensics (allows you to see the steps of an attack and where traffic is going to or coming from)







Wireshark

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Initialization Vector	Length	Info
34	20.476627	VMware_9a:90:0d:00:0c:29	VMware_e9:e5:d1:00:00:00	ARP		60	Who has 192.168.56.2? Tell 192.168.56.128
35	20.476630	VMware_e9:e5:d1:00:00:00	VMware_9a:90:0d:00:0c:29	ARP		60	192.168.56.2 is at 00:50:56:e9:e5:d1
36	20.496982	192.168.56.130	142.250.190.1...	TLSv1.2		93	Application Data
37	20.497499	142.250.190.1...	192.168.56.130	TCP		60	443 → 49666 [ACK] Seq=1 Ack=40 Win=64240 Len=0
38	20.501248	142.250.190.1...	192.168.56.130	TLSv1.2		93	Application Data
39	20.544609	192.168.56.130	142.250.190.1...	TCP		54	49666 → 443 [ACK] Seq=40 Ack=40 Win=65535 Len=0
40	34.084672	192.168.56.128	192.168.56.2	DNS		100	Standard query 0xf9f2 A connectivity-check.ubuntu.com OPT
41	34.086765	192.168.56.2	192.168.56.128	DNS		132	Standard query response 0xf9f2 A connectivity-check.ubuntu.com A 35.2...
42	34.089725	192.168.56.128	35.224.170.84	TCP		74	55562 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=373...
43	35.113870	192.168.56.128	35.224.170.84	TCP		74	[TCP Retransmission] 55562 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460

> Frame 1: 316 bytes on wire (2528 bits), 316 bytes captured (2528 bits)

> Ethernet II, Src: VMware\_9a:90:0d (00:0c:29:9a:90:0d), Dst: IPv4mcast\_02:7f:fe (01:00:5e:02:7f:fe)

> Internet Protocol Version 4, Src: 192.168.56.128, Dst: 224.2.127.254

> User Datagram Protocol, Src Port: 33918, Dst Port: 9875

> Session Announcement Protocol

> Session Description Protocol

0010	01 2e 0f 9f 40 00 ff 11 11 f6 c0 a8 38 80 e0 02	...@... ..8...
0020	7f fe 84 7e 26 93 01 1a 9e 4a 20 00 d5 9e 6c 13	...~&... .J ...l.
0030	0e 10 61 70 70 6c 69 63 61 74 69 6f 6e 2f 73 64	..applic ation/sd
0040	70 00 76 3d 30 0d 0a 6f 3d 2d 20 31 36 35 31 31	p.v=0..o -= 16511
0050	37 37 31 39 35 35 38 31 33 36 38 37 39 39 33 20	77195581 3687993
0060	31 36 35 31 31 37 37 31 39 35 35 38 31 33 36 38	16511771 95581368
0070	37 39 39 33 20 49 4e 20 49 50 34 20 75 62 75 6e	7993 IN IP4 unbun
0080	74 75 2d 32 30 0d 0a 73 3d 41 6e 6e 6f 75 6e 63	tu-20..s =Announc
0090	65 6d 65 6e 74 0d 0a 69 3d 4e 2f 41 0d 0a 63 3d	ement..i =N/A..c=

Packet List

Packet Details

Packet Bytes

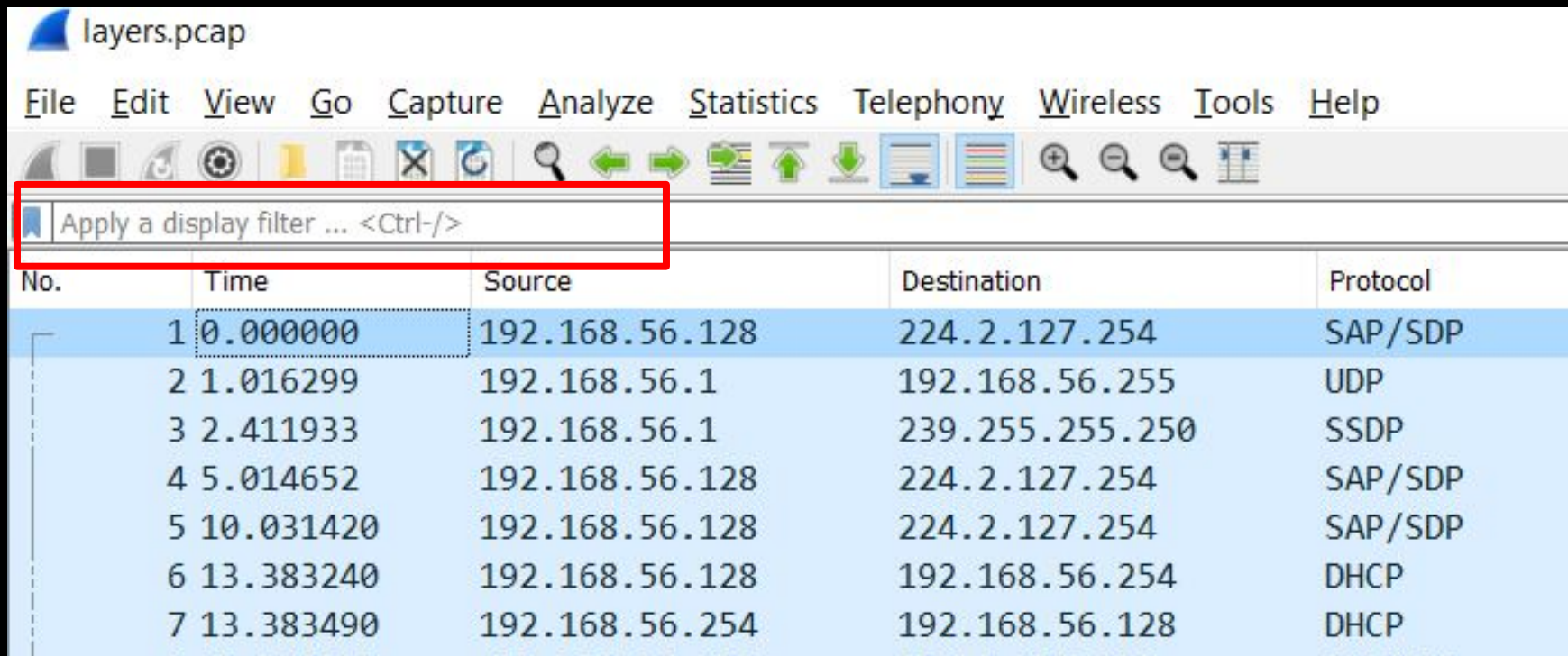
Ready to load or capture

Packets: 1267 · Displayed: 1267 (100.0%)

Profile: Default

# Filters

- Makes analyzing packets so much easier
- Every protocol has its own set of filters to use



No.	Time	Source	Destination	Protocol
1	0.000000	192.168.56.128	224.2.127.254	SAP/SDP
2	1.016299	192.168.56.1	192.168.56.255	UDP
3	2.411933	192.168.56.1	239.255.255.250	SSDP
4	5.014652	192.168.56.128	224.2.127.254	SAP/SDP
5	10.031420	192.168.56.128	224.2.127.254	SAP/SDP
6	13.383240	192.168.56.128	192.168.56.254	DHCP
7	13.383490	192.168.56.254	192.168.56.128	DHCP





# Filtering for HTTP Traffic

layers.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol
→ 175	50.708794	192.168.56.130	192.168.56.128	HTTP
← 177	50.713099	192.168.56.128	192.168.56.130	HTTP
• 185	50.929287	192.168.56.130	192.168.56.128	HTTP
	187 50.930649	192.168.56.128	192.168.56.130	HTTP
	190 50.931936	192.168.56.130	192.168.56.128	HTTP
	194 50.932470	192.168.56.130	192.168.56.128	HTTP
	196 50.933455	192.168.56.128	192.168.56.130	HTTP
	198 50.933633	192.168.56.128	192.168.56.130	HTTP
	200 50.934247	192.168.56.130	192.168.56.128	HTTP



# Filtering for IP Address

layers.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr==192.168.56.1

No.	Time	Source	Destination	Protocol
2	1.016299	192.168.56.1	192.168.56.255	UDP
3	2.411933	192.168.56.1	239.255.255.250	SSDP
148	45.761374	192.168.56.1	192.168.56.255	UDP
728	68.001826	192.168.56.1	239.255.255.250	SSDP
729	68.112693	192.168.56.1	239.255.255.250	SSDP
730	69.003739	192.168.56.1	239.255.255.250	SSDP
731	69.115517	192.168.56.1	239.255.255.250	SSDP
732	70.003766	192.168.56.1	239.255.255.250	SSDP
733	70.114809	192.168.56.1	239.255.255.250	SSDP





# Isolating Conversations/Streams

- There are a lot of different conversations and streams that can be present in a single packet capture
- Sometimes, it is better to view only one conversation at a time
- Filter examples:
  - `tcp.stream==15`
  - `udp.stream==1`

The screenshot shows the Wireshark interface with a packet capture list on the left. Packet 161 is selected, and a context menu is open over it. The menu options are:

- Mark/Unmark Packet (Ctrl+M)
- Ignore/Unignore Packet (Ctrl+D)
- Set/Unset Time Reference (Ctrl+T)
- Time Shift... (Ctrl+Shift+T)
- Packet Comment... (Ctrl+Alt+C)
- Edit Resolved Name
- Apply as Filter
- Prepare as Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow (highlighted)
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

The 'Follow' submenu is open, showing the following options:

- TCP Stream (Ctrl+Alt+Shift+T)
- UDP Stream (Ctrl+Alt+Shift+U)
- TLS Stream (Ctrl+Alt+Shift+S)
- HTTP Stream (Ctrl+Alt+Shift+H)
- HTTP/2 Stream
- QUIC Stream

The packet list shows the following details for packet 161:

No.	Time	Source	Destination	Protocol	Length
155	18.784093	192.168.56.132	192.168.56.2	DNS	77
156	18.785915	192.168.56.2	192.168.56.132	DNS	373
157	18.787214	192.168.56.132	72.21.91.29	TCP	66
158	18.788511	192.168.56.132	72.21.91.29	TCP	66
159	18.791993	72.21.91.29	192.168.56.132	TCP	60
160	18.792268	192.168.56.132	72.21.91.29	TCP	60
161	18.792662	192.168.56.132	72.21.91.29	HTTP	285
162	18.792662	192.168.56.132	72.21.91.29	TCP	60
163	18.792662	192.168.56.132	72.21.91.29	TCP	60
164	18.792662	192.168.56.132	72.21.91.29	TCP	60
165	18.792662	192.168.56.132	72.21.91.29	HTTP	289
166	18.792662	192.168.56.132	72.21.91.29	TCP	60

The packet details pane shows the following information for packet 161:

- Frame 161: 285 bytes on wire (2280 bits) captured (0.000000000 seconds) on interface eth0
- Ethernet II, Src: Intel E81C8 (82:55:c8:00:08:1c), Dst: Intel E81C8 (82:55:c8:00:08:1c)
- Internet Protocol Version 4, Src: 192.168.56.132, Dst: 72.21.91.29
- Transmission Control Protocol, Seq: 373500000, Win: 65535, Len: 0
- Hypertext Transfer Protocol, GET / HTTP/1.1

The packet bytes pane shows the raw data in hexadecimal and ASCII:

Offset	Hex	ASCII
0	0000	
1	0010	
2	0020	
3	0030	
4	0040	
5	0050	
6	0060	
7	0070	
8	0080	
9	0090	
10	00a0	
11	00b0	
12	00c0	
13	00d0	
14	00e0	
15	00f0	
16	0100	
17	0110	
18	0120	
19	0130	
20	0140	
21	0150	
22	0160	
23	0170	
24	0180	
25	0190	
26	01a0	
27	01b0	
28	01c0	
29	01d0	
30	01e0	
31	01f0	
32	0200	
33	0210	
34	0220	
35	0230	
36	0240	
37	0250	
38	0260	
39	0270	
40	0280	
41	0290	
42	02a0	
43	02b0	
44	02c0	
45	02d0	
46	02e0	
47	02f0	
48	0300	
49	0310	
50	0320	
51	0330	
52	0340	
53	0350	
54	0360	
55	0370	
56	0380	
57	0390	
58	03a0	
59	03b0	
60	03c0	
61	03d0	
62	03e0	
63	03f0	
64	0400	
65	0410	
66	0420	
67	0430	
68	0440	
69	0450	
70	0460	
71	0470	
72	0480	
73	0490	
74	04a0	
75	04b0	
76	04c0	
77	04d0	
78	04e0	
79	04f0	
80	0500	
81	0510	
82	0520	
83	0530	
84	0540	
85	0550	
86	0560	
87	0570	
88	0580	
89	0590	
90	05a0	
91	05b0	
92	05c0	
93	05d0	
94	05e0	
95	05f0	
96	0600	
97	0610	
98	0620	
99	0630	
100	0640	
101	0650	
102	0660	
103	0670	
104	0680	
105	0690	
106	06a0	
107	06b0	
108	06c0	
109	06d0	
110	06e0	
111	06f0	
112	0700	
113	0710	
114	0720	
115	0730	
116	0740	
117	0750	
118	0760	
119	0770	
120	0780	
121	0790	
122	07a0	
123	07b0	
124	07c0	
125	07d0	
126	07e0	
127	07f0	
128	0800	
129	0810	
130	0820	
131	0830	
132	0840	
133	0850	
134	0860	
135	0870	
136	0880	
137	0890	
138	08a0	
139	08b0	
140	08c0	
141	08d0	
142	08e0	
143	08f0	
144	0900	
145	0910	
146	0920	
147	0930	
148	0940	
149	0950	
150	0960	
151	0970	
152	0980	
153	0990	
154	09a0	
155	09b0	
156	09c0	
157	09d0	
158	09e0	
159	09f0	
160	0a00	
161	0a10	
162	0a20	
163	0a30	
164	0a40	
165	0a50	
166	0a60	
167	0a70	
168	0a80	
169	0a90	
170	0aa0	
171	0ab0	
172	0ac0	
173	0ad0	
174	0ae0	
175	0af0	
176	0b00	
177	0b10	
178	0b20	
179	0b30	
180	0b40	
181	0b50	
182	0b60	
183	0b70	
184	0b80	
185	0b90	
186	0ba0	
187	0bb0	
188	0bc0	
189	0bd0	
190	0be0	
191	0bf0	
192	0c00	
193	0c10	
194	0c20	
195	0c30	
196	0c40	
197	0c50	
198	0c60	
199	0c70	
200	0c80	
201	0c90	
202	0ca0	
203	0cb0	
204	0cc0	
205	0cd0	
206	0ce0	
207	0cf0	
208	0d00	
209	0d10	
210	0d20	
211	0d30	
212	0d40	
213	0d50	
214	0d60	
215	0d70	
216	0d80	
217	0d90	
218	0da0	
219	0db0	
220	0dc0	
221	0dd0	
222	0de0	
223	0df0	
224	0e00	
225	0e10	
226	0e20	
227	0e30	
228	0e40	
229	0e50	
230	0e60	
231	0e70	
232	0e80	
233	0e90	
234	0ea0	
235	0eb0	
236	0ec0	
237	0ed0	
238	0ee0	
239	0ef0	
240	0f00	
241	0f10	
242	0f20	
243	0f30	
244	0f40	
245	0f50	
246	0f60	
247	0f70	
248	0f80	
249	0f90	
250	0fa0	
251	0fb0	
252	0fc0	
253	0fd0	
254	0fe0	
255	0ff0	

# Wireshark in Scripting and CLI

- tcpdump: create a packet capture
- tshark: extract data from a packet capture
- PyShark: Python wrapper for tshark to use in scripts





# Burp Suite

- Proxy tool to MITM your own web traffic
- Why? To modify requests to the web application and try to break it
- Like Wireshark, but made specifically to attack web applications

The screenshot displays the Burp Suite interface, version 2021.9.1. The top menu bar includes options like Burp, Project, Intruder, Repeater, Window, Help, and Burp Suite Community Edition. Below the menu, there are tabs for Decoder, Comparer, Logger, Extender, Project options, User options, and Learn. The main interface is divided into several sections: Dashboard, Target, Proxy (selected), Intruder, Repeater, and Sequencer. The Proxy tab is active, showing a list of intercepted HTTP requests. The filter is set to 'Hiding CSS, image and general binary content'. The table below shows the intercepted requests:

#	Host	Method	URL	Params	Edited	Status
1	https://www.google.com	GET	/			200
2	https://www.google.com	POST	/gen_204?atyp=i&ei=QJ6NYcXyKfyk2r...	✓		204
3	https://www.gstatic.com	GET	/og/_js/k=og.qtm.en_US.75zE3OGOif4...			304
4	https://www.google.com	GET	/images/searchbox/desktop_searchbo...			304
5	https://www.google.com	GET	/logos/doodles/2021/veterans-day-20...			304
6	https://www.google.com	GET	/xjs/_js/k=xjs.s.en_US.OaUGqFwxXok...			304
7	https://www.google.com	POST	/gen_204?s=webhp&t=aft&atyp=csi&...	✓		204
8	https://token.services.mozilla.co...	GET	/1.0/sync/1.5			401
9	https://apis.google.com	GET	/_scs/abc-static/_js/k=gapi.gapi.en.R...			304
10	https://www.google.com	GET	/complete/search?q&cp=0&client=gw...	✓		200
11	https://www.google.com	GET	/xjs/_js/k=xjs.s.en_US.OaUGqFwxXok...	✓		304
12	https://www.google.com	GET	/client_204?&atyp=i&biw=1536&bih=7...	✓		204

Below the table, the 'Request' and 'Response' tabs are visible. The 'Request' tab is selected, showing the raw HTTP request details for the first entry (GET / HTTP/1.1). The 'Response' tab is also visible, showing the raw HTTP response details (HTTP/2 200 OK). The 'INSPECTOR' panel on the right shows the request attributes, cookies, headers, and response headers.

# Challenges

**Layers 1-7:** easy, approachable Wireshark challenges teaching OSI

**File Transfer:** analyzing FTP data traffic (layer 7)

**Pool:** using filters effectively to isolate traffic (layers 5-7)

**Livestream Fail:** extracting video stream (layer 6)

**toobeetootee:** analyzing Minetest game traffic (layers 6-7)

- Note: this challenge was part of UIUCTF 2021, please avoid writeups related to the challenge



# Next Meetings

## **Weekend Seminar:** Wireless Networking

- How to break into wireless networks

## **Thursday:** Windows Environments

- Talking about hell i mean hell i mean hell i mean windows
- Active Directory, Windows systems, Domain controllers, NTLM, SMB etc

