

# Week 11

# Networking

Minh Duong



# Announcements

Shib auth, we are in need of maintainer/s

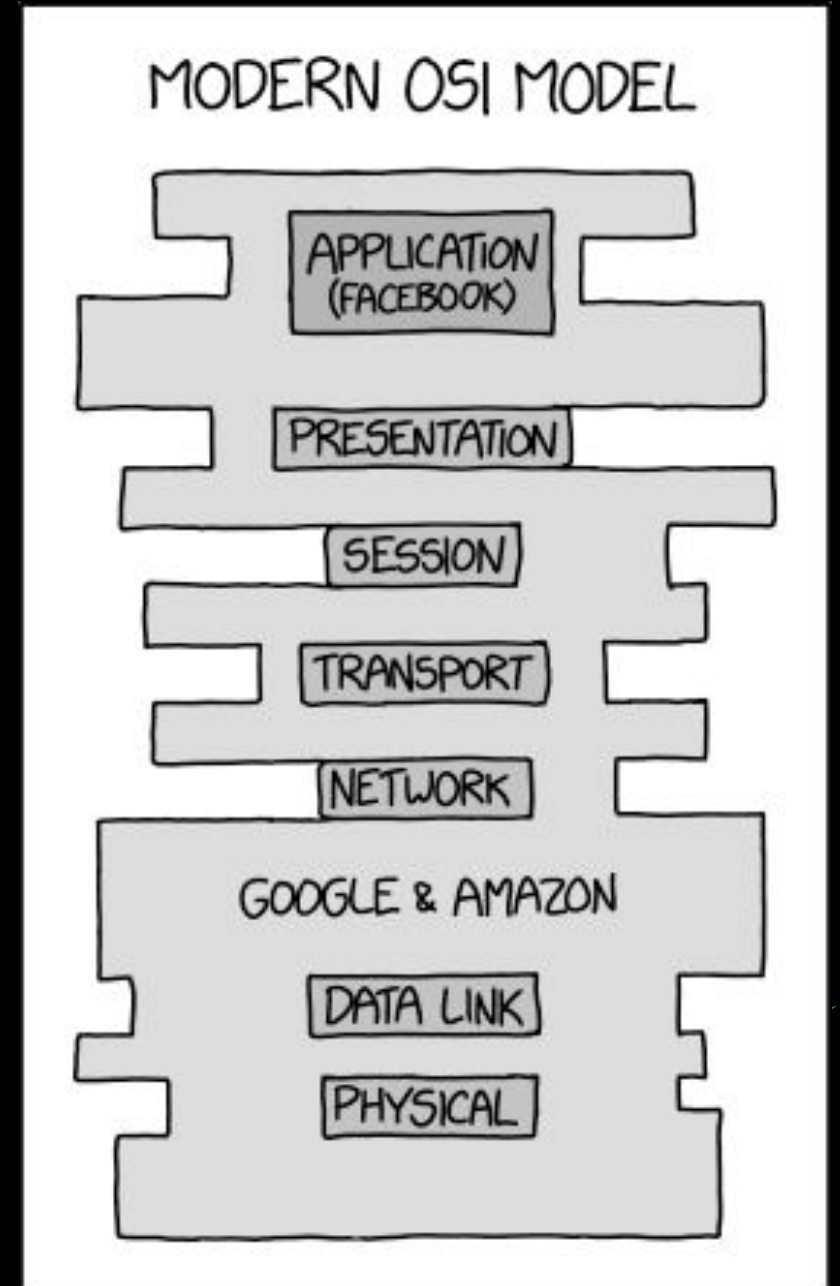
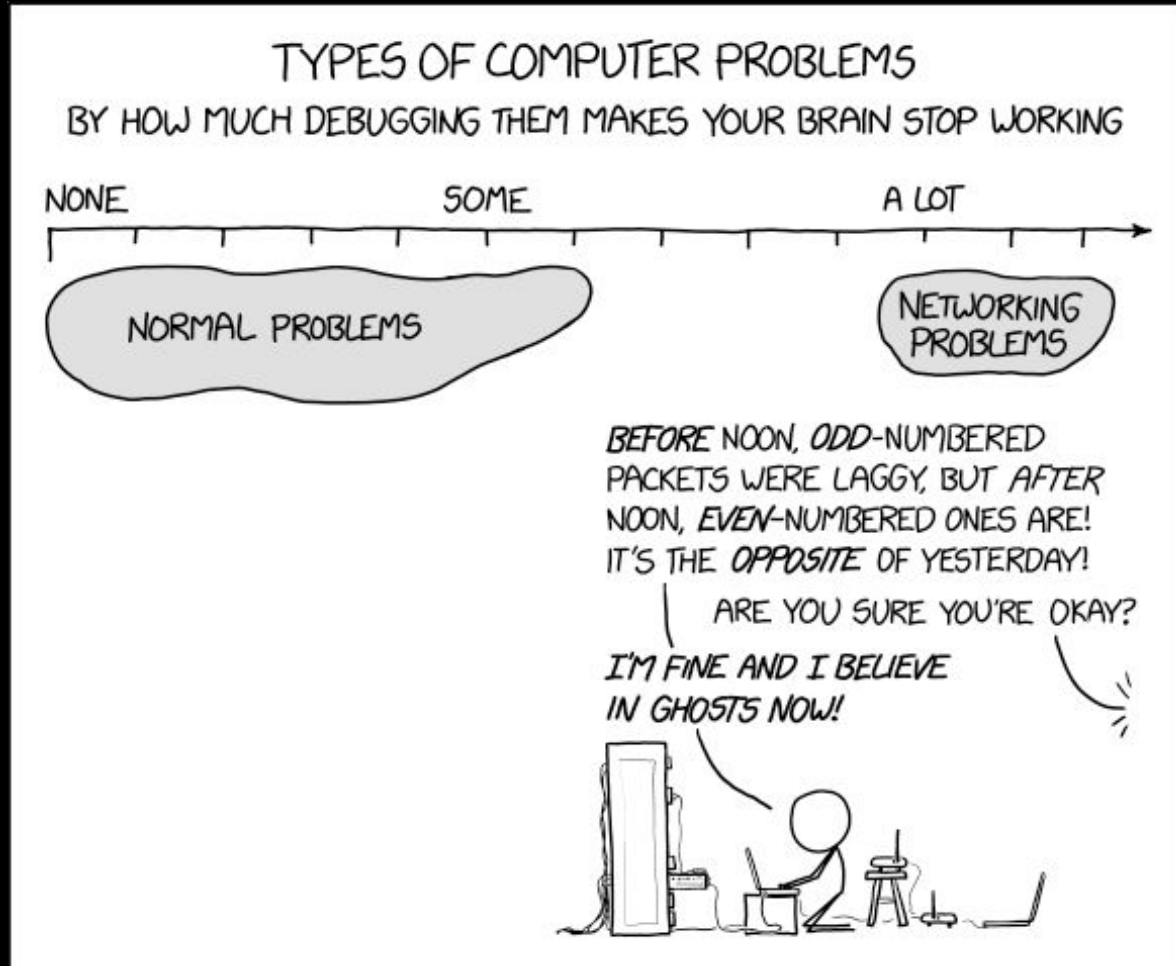
Website: we also need maintainers

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

Spray paint social @ some point

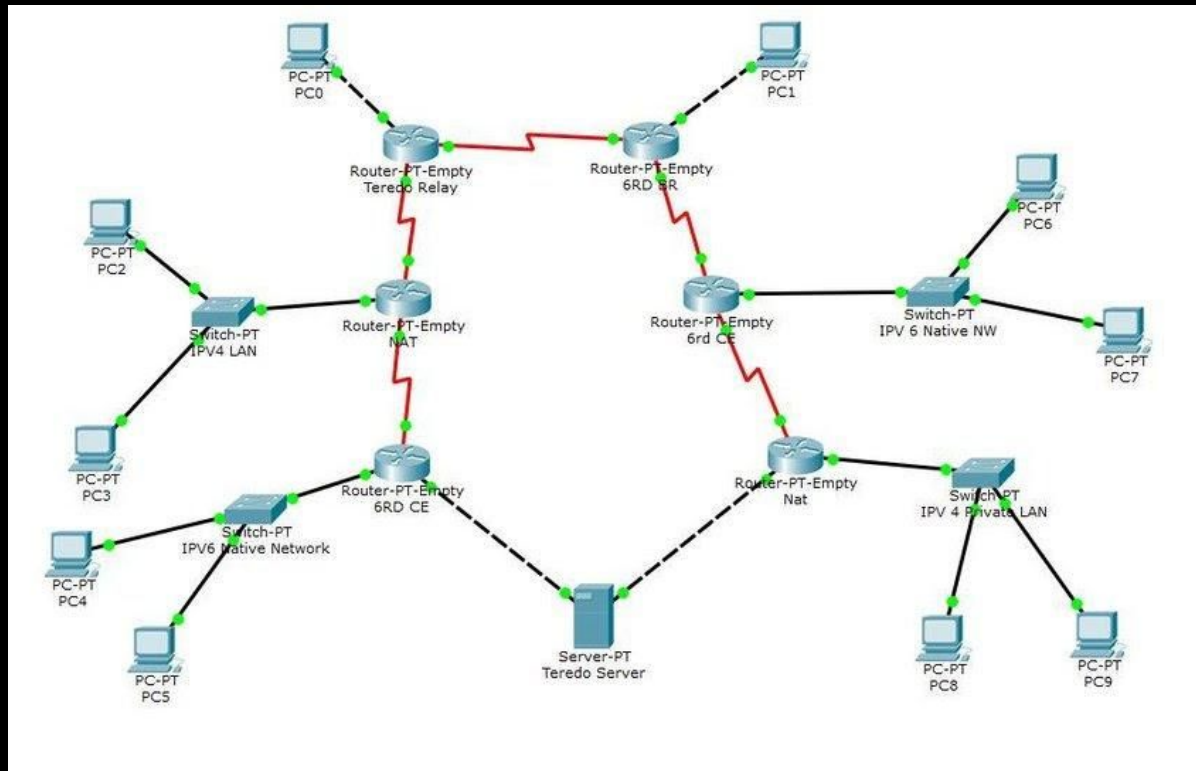


```
sigpwny{please_do_
not_throw_sausage_
pizza_away}
```



# 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

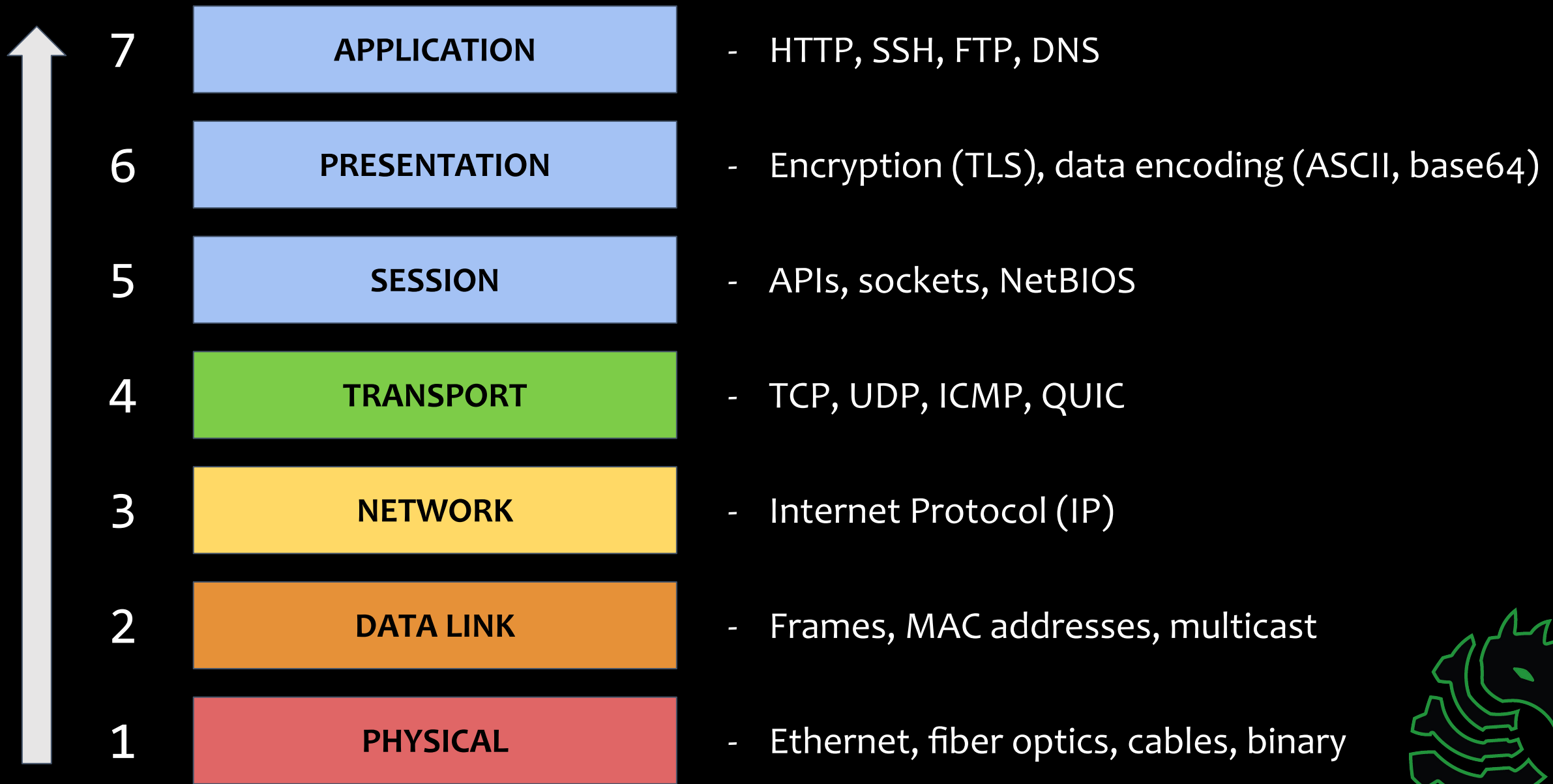
- If devices all speak different networking languages, then they can't understand each other
- As a result, protocols and standards are needed
- There are lots of networking protocols... and a lot of acronyms



# 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)

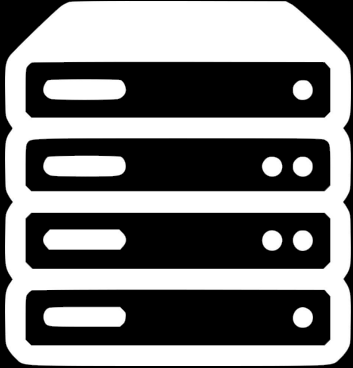
A



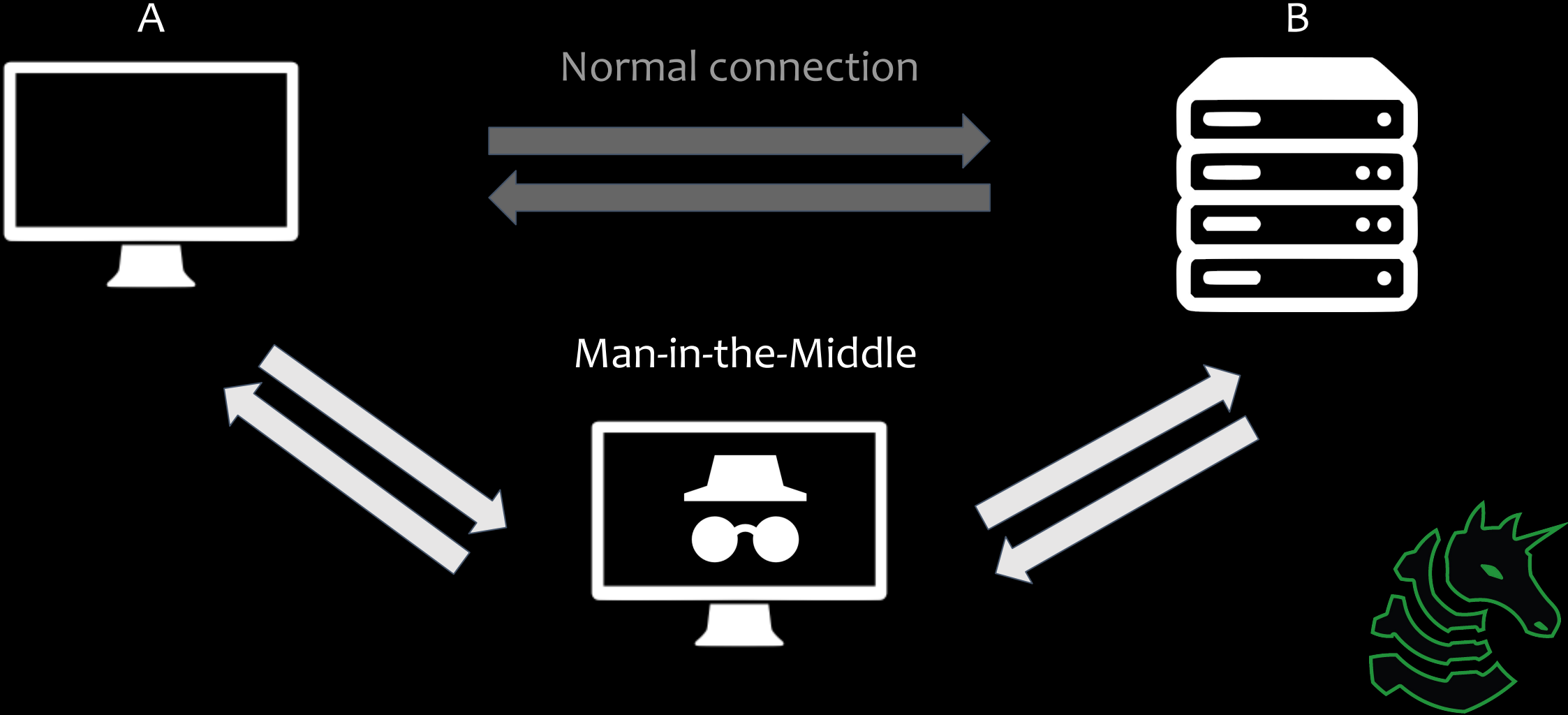
Normal connection



B



# 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)





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

No.	Time	Source	Destination	Protocol	Initialization Vector	Length	Info
34	20.476627	VMware_9a:90:...	VMware_e9:e5:...	ARP		60	Who has 192.168.56.2? Tell 192.168.56.128
35	20.476630	VMware_e9:e5:...	VMware_9a:90:...	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	35.113979	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

```

0000  01 00 5e 02 7f fe 00 0c 29 9a 90 0d 08 00 45 00  ..^.....).....E.
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 ubun
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=

```

No.	Time	Source	Destination	Protocol	Initialization Vector	Length	Info
34	20.476627	VMware_9a:90:...	VMware_e9:e5:...	ARP		60	Who has 192.168.56.2? Tell 192.168.56.128
35	20.476630	VMware_e9:e5:...	VMware_9a:90:...	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	35.113070	192.168.56.128	35.224.170.84	TCP		74	[TCP Retransmission] 55562 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460

## Packet List

```

> 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

```

## Packet Details

```

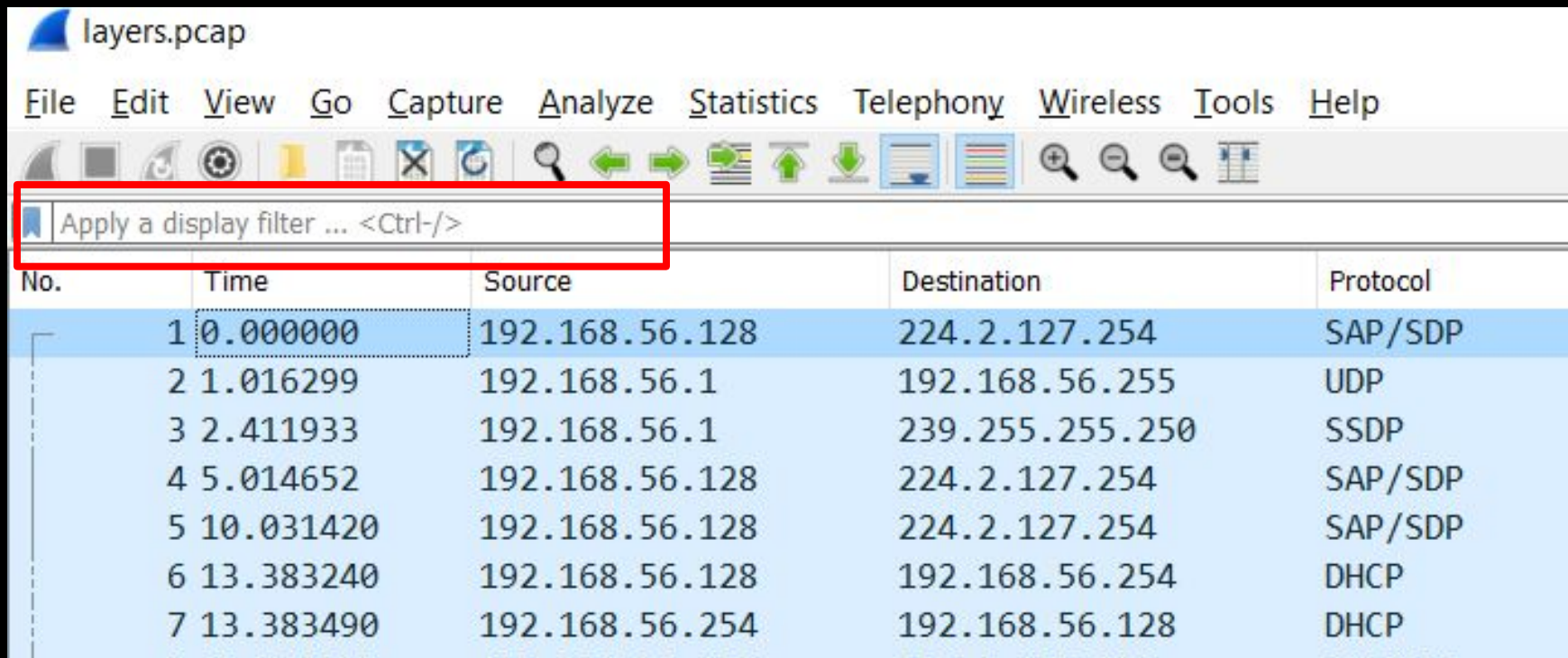
0010  01 2e 0f 9f 40 00 ff 11 11 f6 c0 a8 38 80 e0 02  . . . @ . . . . . 8 . . .
0020  7f fe 84 7e 26 93 01 1a 9e 4a 20 00 d5 9e 6c 13  . . . ~ & . . . . J . . . l .
0030  0e 10 61 70 70 6c 69 63 61 74 69 6f 6e 2f 73 64  . . applic ation/sd
0040  70 00 76 3d 30 0d 0a 6f 3d 2d 20 31 36 35 31 31  p . v = 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 ubun
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 Bytes

# Filters

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

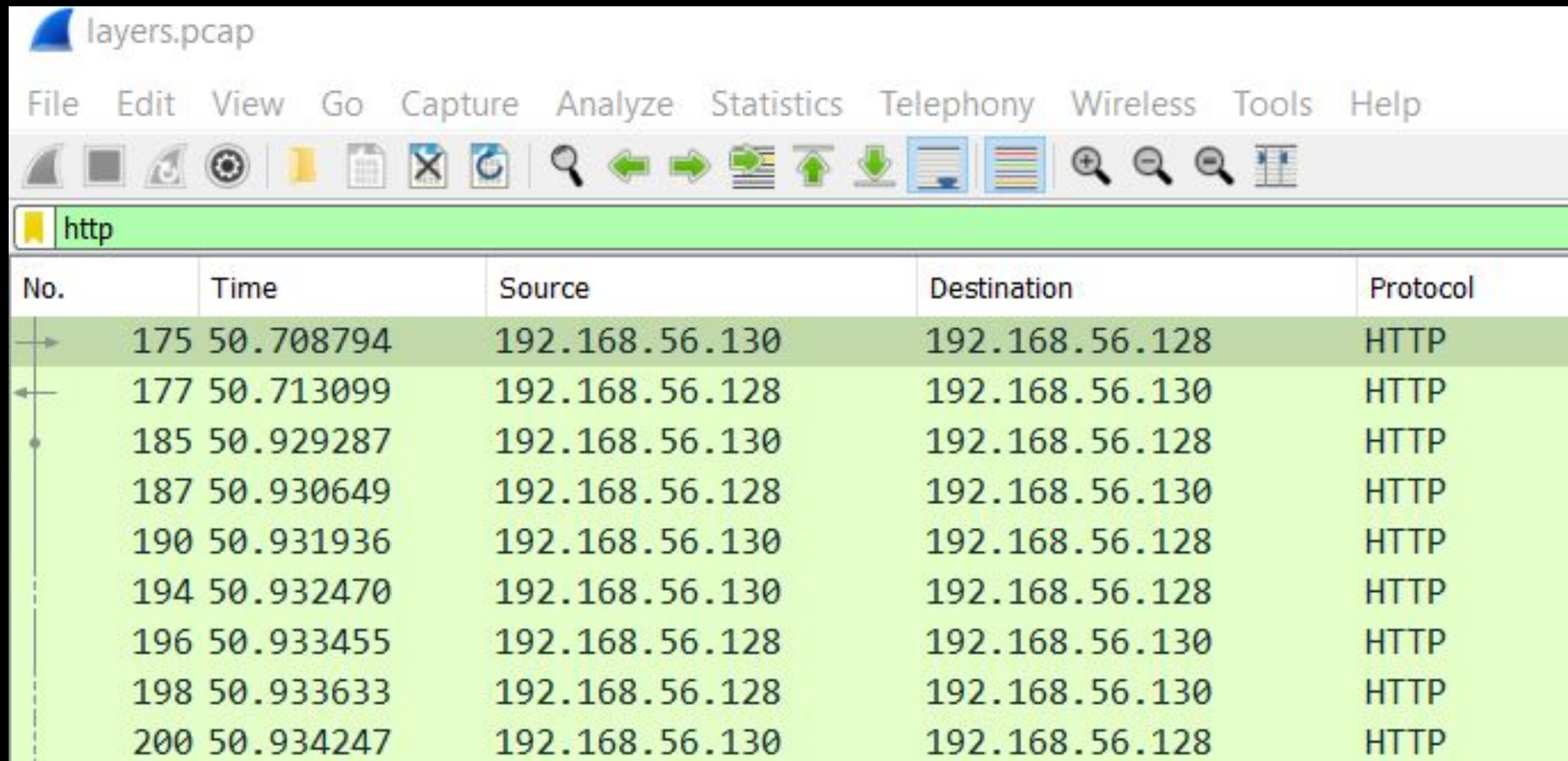


The screenshot shows the Wireshark interface with a packet capture named 'layers.pcap'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for navigation and analysis. A red box highlights the 'Apply a display filter ... <Ctrl-/>' input field. Below the toolbar is a table of captured packets.

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



The screenshot shows the Wireshark interface with a filter applied to the packet list. The filter bar at the top contains the text "http". The packet list below shows several packets, all of which are HTTP traffic. The columns are labeled "No.", "Time", "Source", "Destination", and "Protocol".

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 list on the left and a packet details pane on the right. A context menu is open over packet 161, which is an HTTP packet. The 'Follow' option is selected, and a sub-menu is visible showing various stream types: 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, and QUIC Stream.

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	18.793056	192.168.56.132	72.21.91.29	TCP	60
163	18.793450	192.168.56.132	72.21.91.29	TCP	60
164	18.793844	72.21.91.29	192.168.56.132	TCP	60
165	18.794238	72.21.91.29	192.168.56.132	HTTP	289
166	18.794632	192.168.56.132	72.21.91.29	TCP	60

Frame	Ethernet II	Internet Protocol Version 4	Transmission Control Protocol	Hypertext Transfer Protocol
bits), 285 bytes	0000 00 50 56 e9 e5 d1	(00:0c:29:22:5	0010 01 0f 7d b5 40 00	0020 5b 1d e4 e8 00 50
192.168.56.132	0030 fa f0 4e 5f 00 00	Port: 58600,	0040 7a 42 4e 4d 45 73	
				4d 43 47 67 55
				45 4a 51 5a 50
				56 59 6f 77 51
				53 42 31 4a 67
				4d 43 45 41 54
				41 72 51 72 68
				00 2f 31 2e 31

# 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. The top menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', 'Help', 'Burp Suite Community Edition v2021.9.1 - Te...', and window control icons. Below the menu is a toolbar with tabs for 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. A secondary toolbar contains 'Dashboard', 'Target', 'Proxy' (highlighted), 'Intruder', 'Repeater', and 'Sequencer'. The main area shows the 'HTTP history' tab with a filter: 'Filter: Hiding CSS, image and general binary content'. A table lists 12 HTTP requests to various Google domains. The bottom section is split into 'Request' and 'Response' views, with a 'Raw' view selected for both. The 'Request' view shows a GET request for '/ HTTP/1.1' with various cookies and headers. The 'Response' view shows an HTTP/2 200 OK response with headers like 'Date: Thu, 11 Nov 2021', 'Expires: -1', 'Cache-Control: private', 'Content-Type: text/html', 'Strict-Transport-Security', 'Server: gws', 'Content-Length: 112635', 'X-Xss-Protection: 0', and 'X-Frame-Options: SAMEORIGIN'. To the right of the request and response views is the 'INSPECTOR' panel, which is currently empty.

#	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

```
1 GET / HTTP/1.1
2 Host: www.google.com
3 Cookie: 1P_JAR=
  2021-11-11-22; NID=
  511=s4lugm-EqFNnNLs6UW1
  bW1YdwJOJ5-3sQ5xnSi8-6B
  klhouaZYGvviJzPYUIJasRC
  kEThIqimxQkS9AyTpVhCKg0
  06Nplq38W00wRjyW50L8-eC
  7zbZL58sV0R46pAU949tmYx
4
5
6
7
8
9
10
```

```
1 HTTP/2 200 OK
2 Date: Thu, 11 Nov 2021
3 Expires: -1
4 Cache-Control: private,
5 Content-Type: text/html
6 Strict-Transport-Security
7 Server: gws
8 Content-Length: 112635
9 X-Xss-Protection: 0
10 X-Frame-Options: SAMEORIGIN
```

# 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

