SIGPwny

Purple Team FA2026 • 2026-09-08

# Network Security & Active Recon

Krishnan Shankar, Pranav Srinivasan

# Krishnan Shankar

- SIGPwny Admin
- Computer Engineering ‘27
- Fun fact: I built my own bike

# Pranav Srinivasan

- SigPwny Purple Team Lead
- Computer Engineering ‘28
- Fun fact: Can kinda rival Ronan’s cursed course credit history

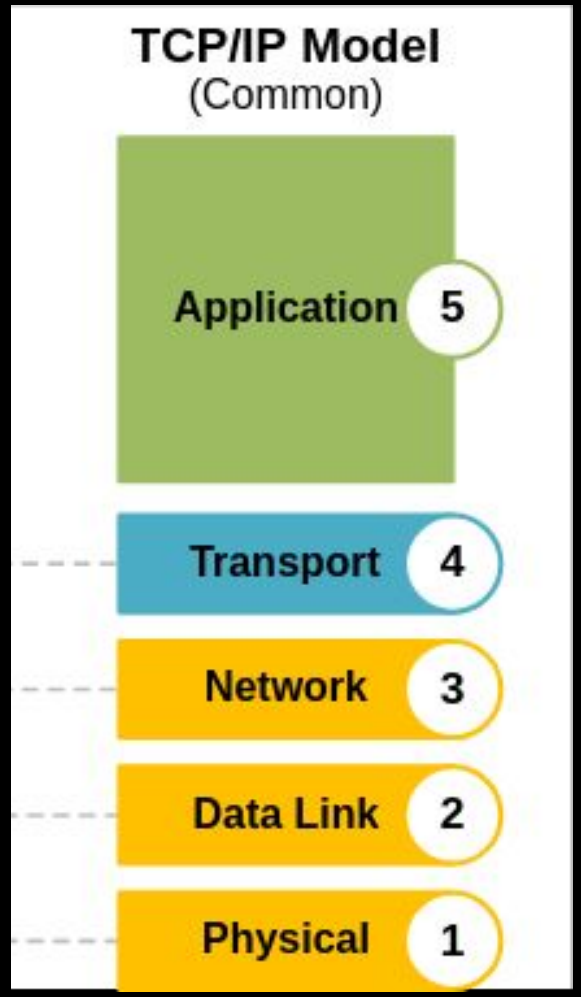
# Table of Contents



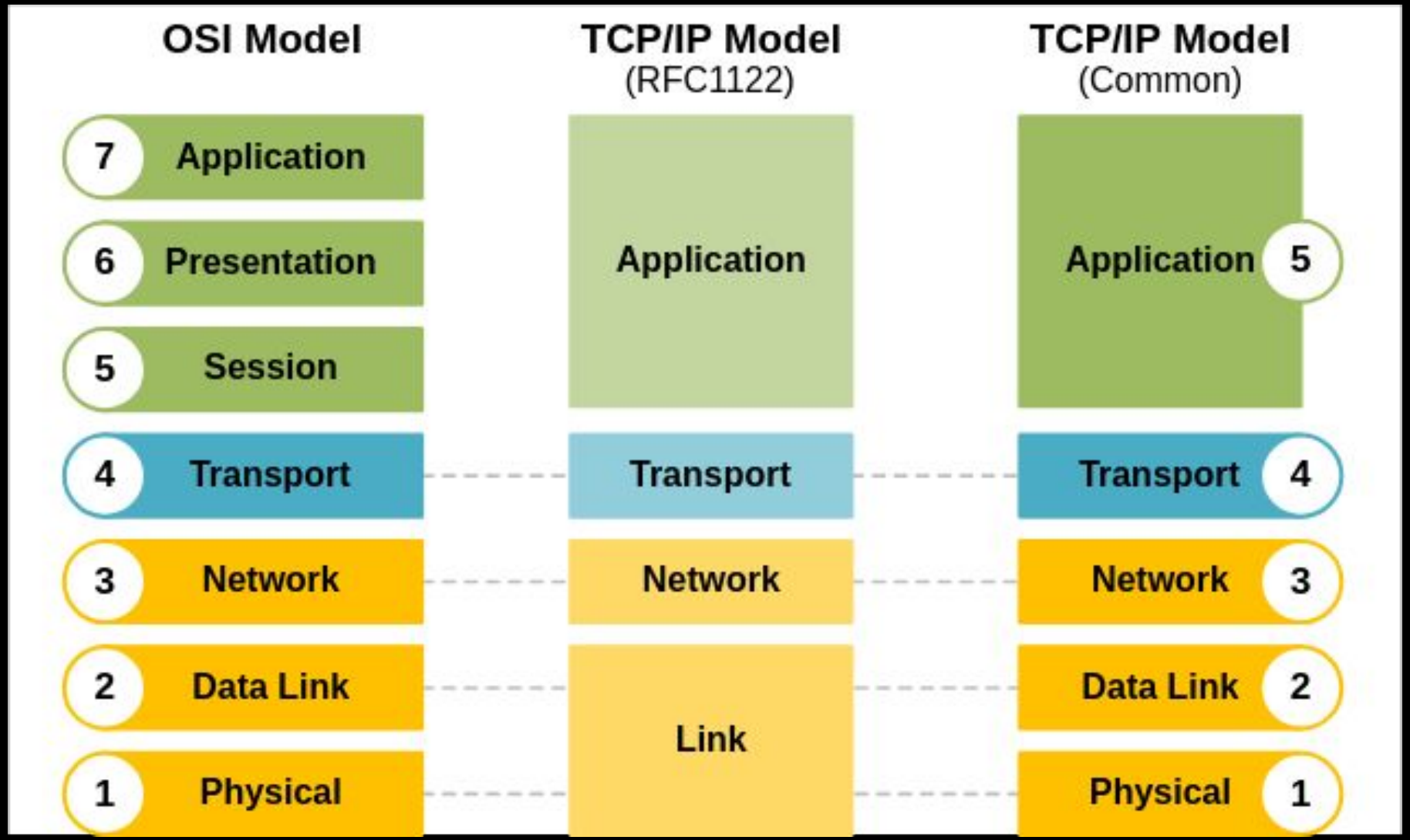
# Networking Basics

# TCP/IP model

- Networking is fundamentally composed of “layers”
- The TCP/IP model offers a very simplified view of these layers
- In purple team, our focus is on the Transport and Application layers
- We will cover the other layers briefly

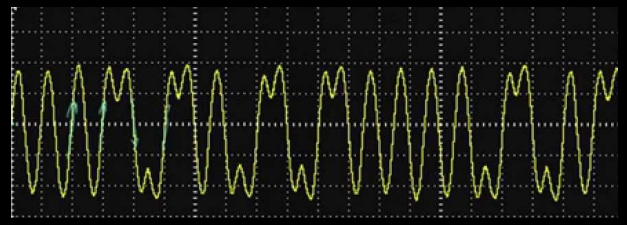


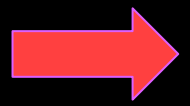

# The OSI Model

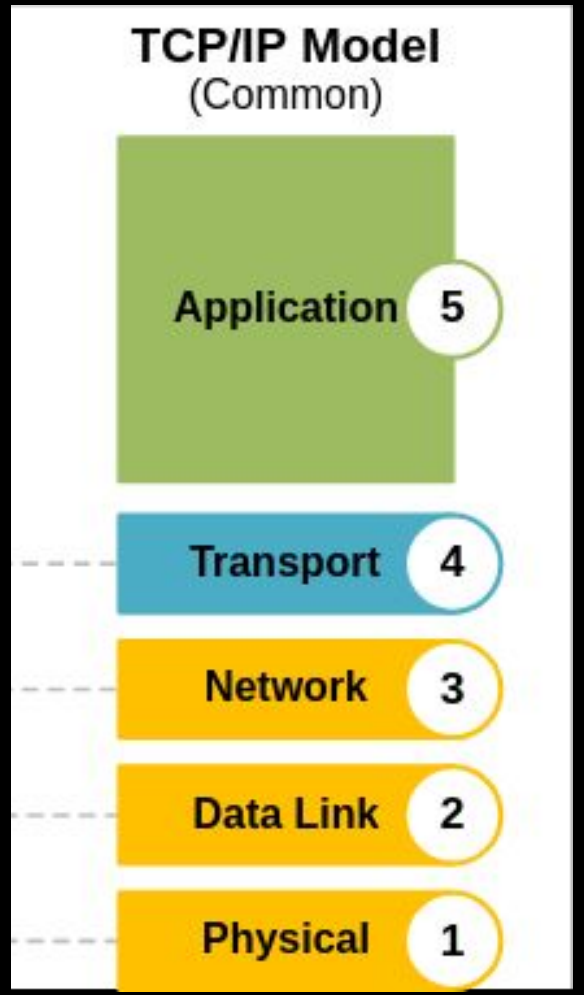


# Physical Layer

- How do raw bits get transmitted?

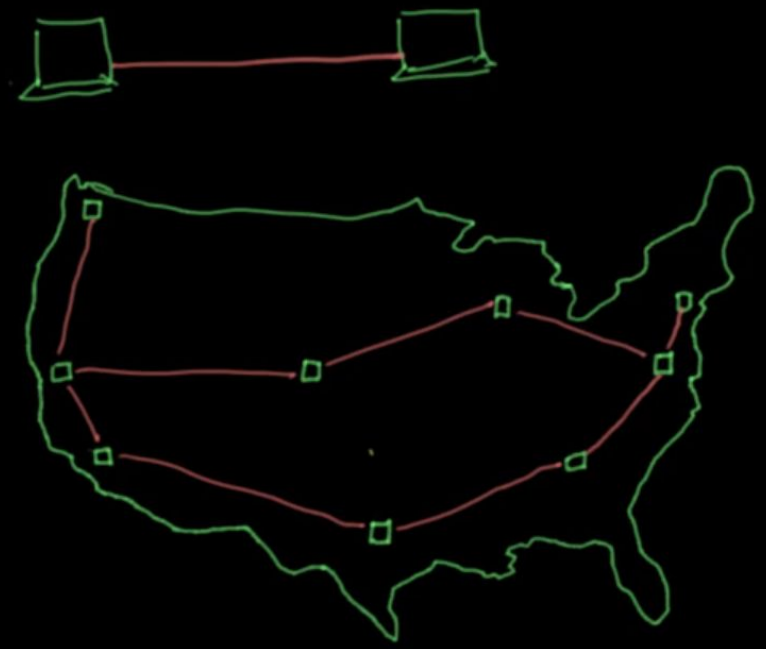


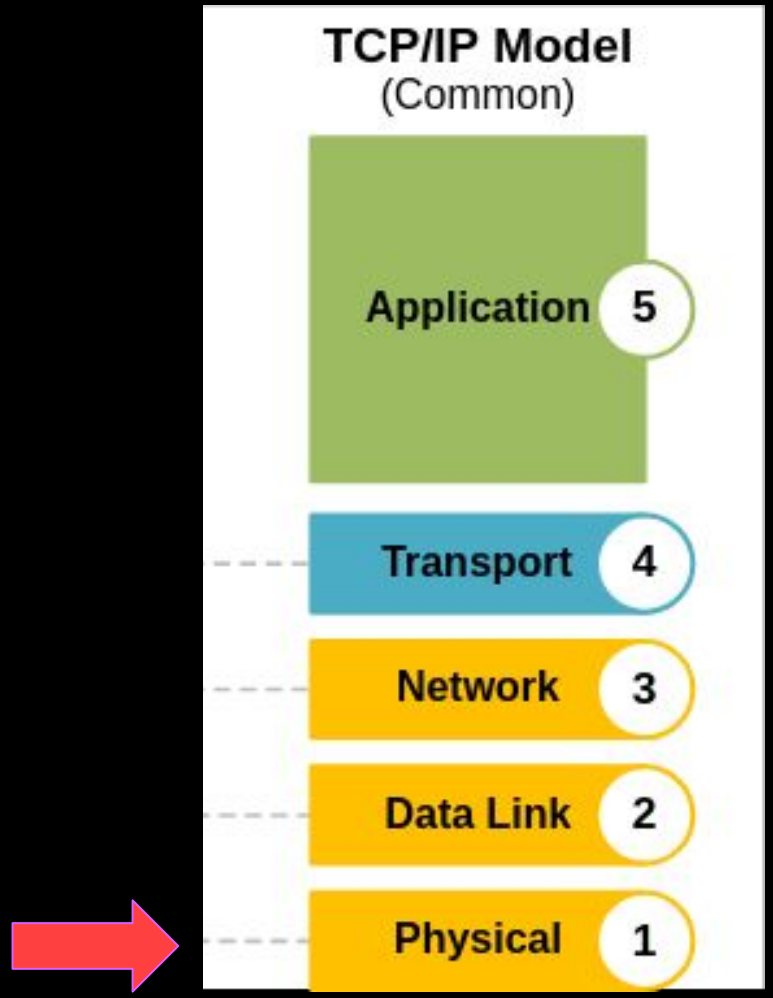





# Physical Layer

- This gives us point-to-point communication of bits

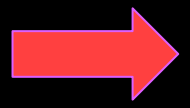


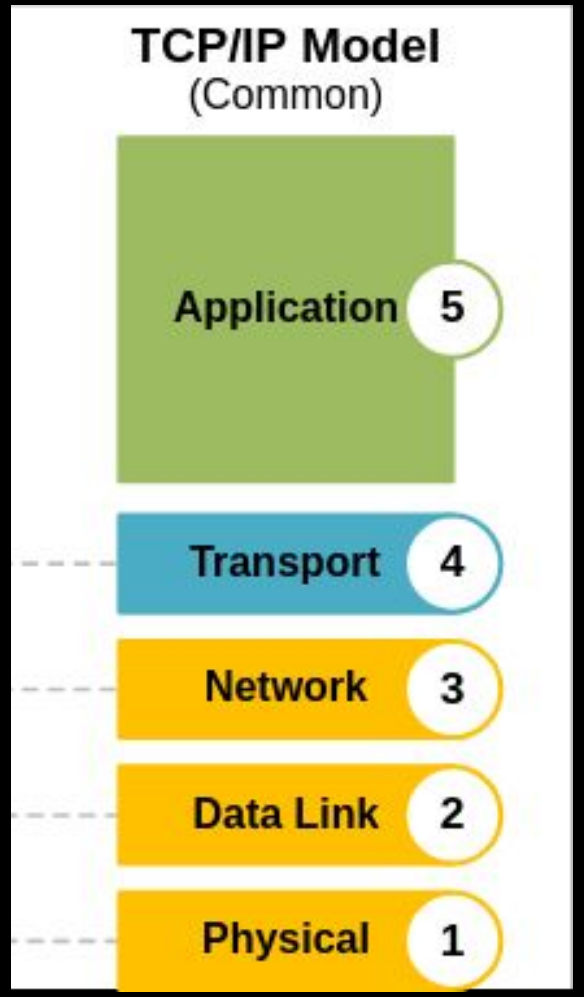



Data

# Physical Layer

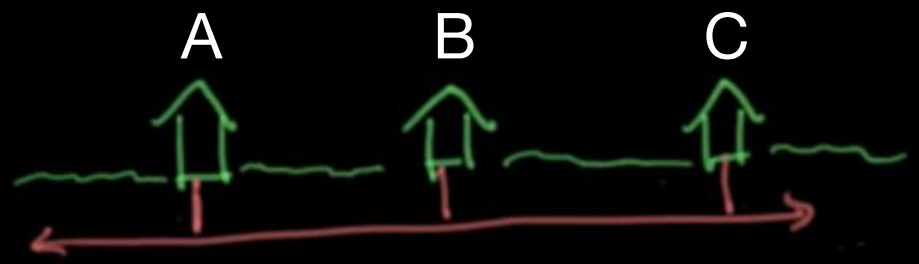
- This gives us point-to-point communication
- But what if we want to re-use the same data line?

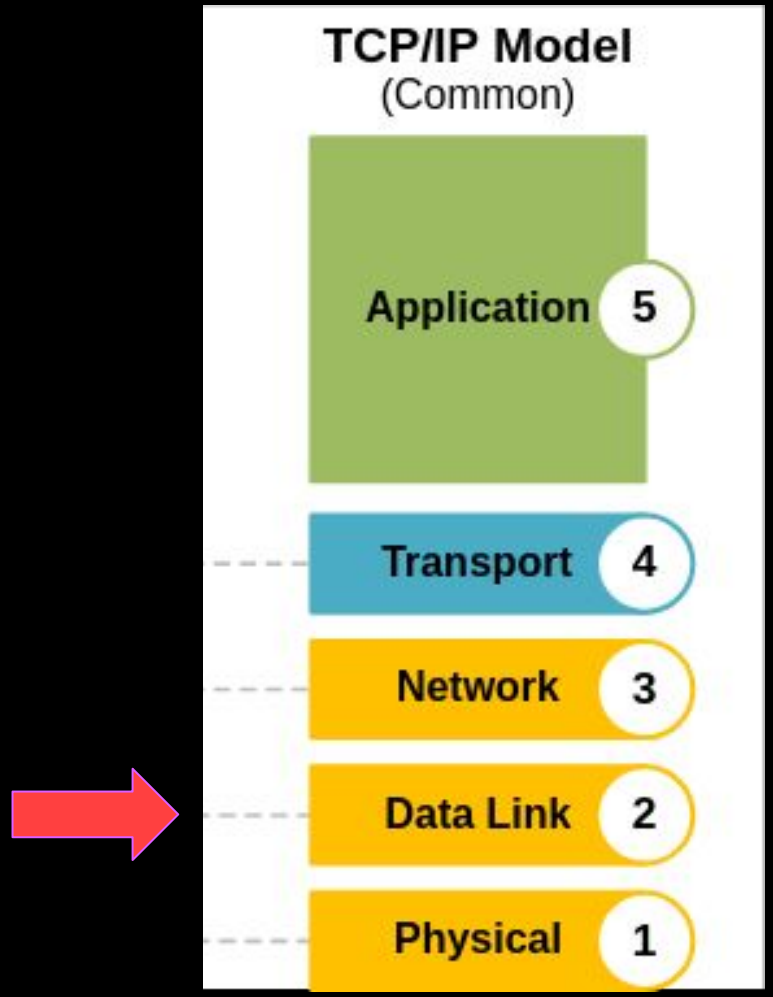






# Data Link Layer

- This gives us point-to-point communication
- But what if we want to re-use the same data line?
- Give every device an address

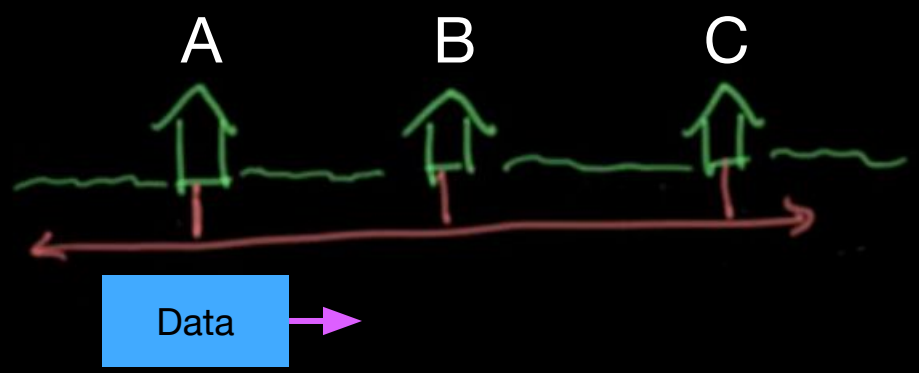


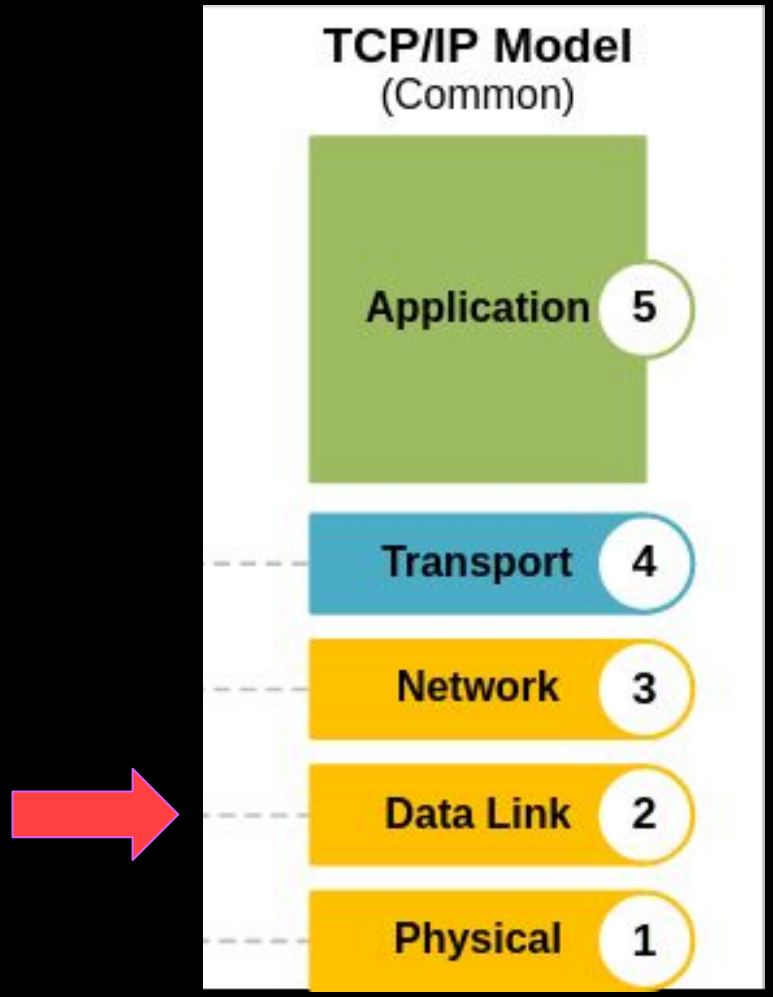




Data

# Data Link Layer

- Give every device an address

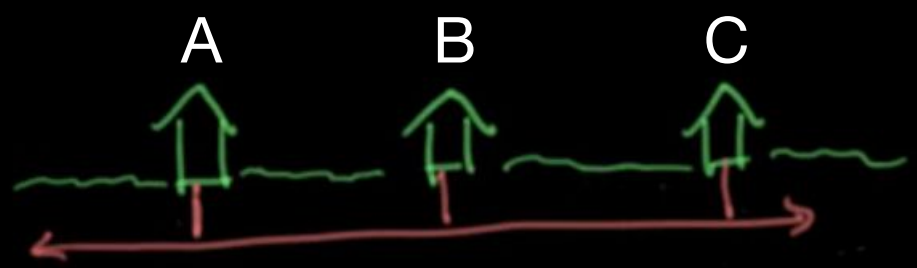


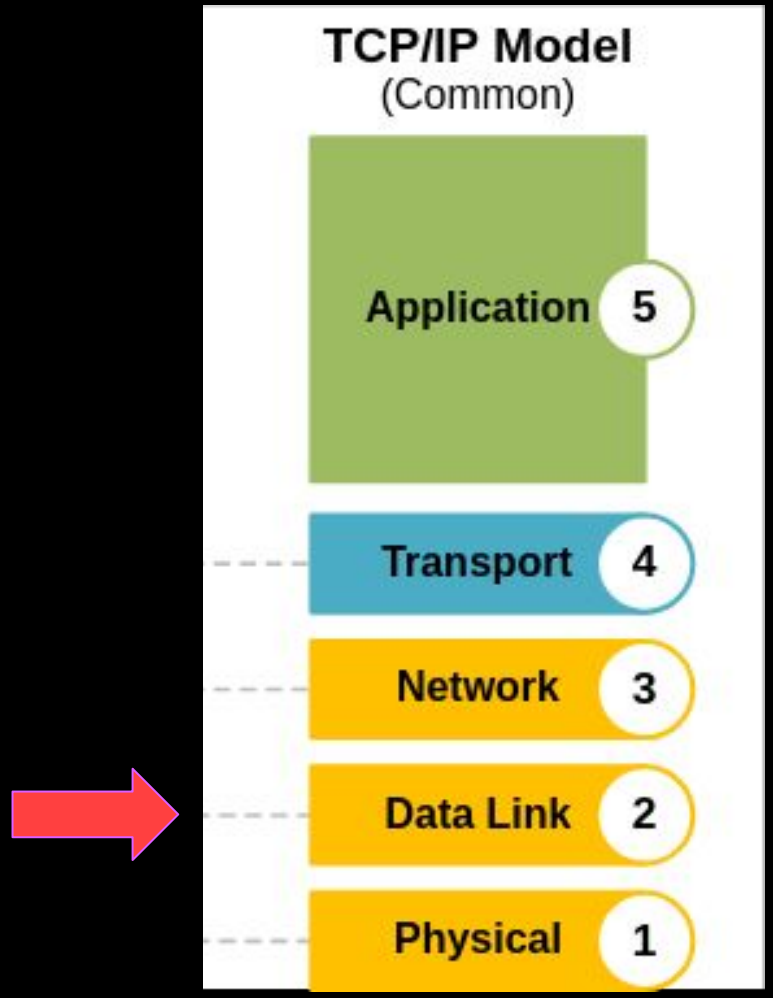




Data

# Data Link Layer

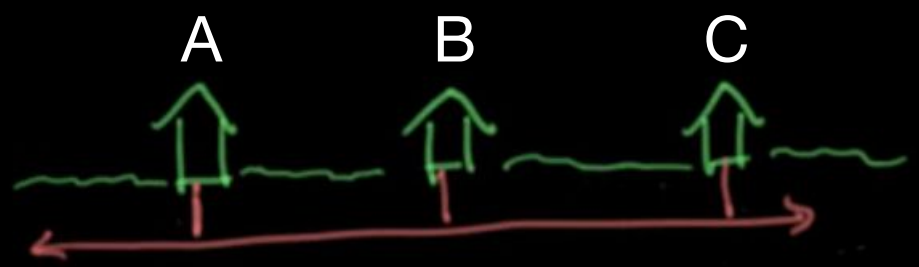
- Give every device an address



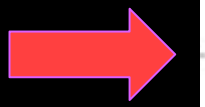



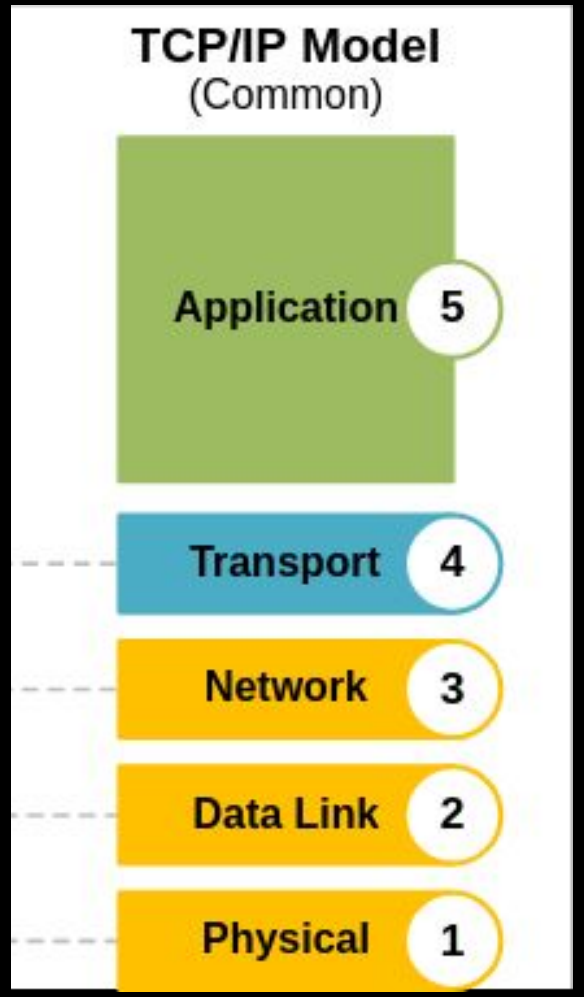

# Data Link Layer

- Give every device an address





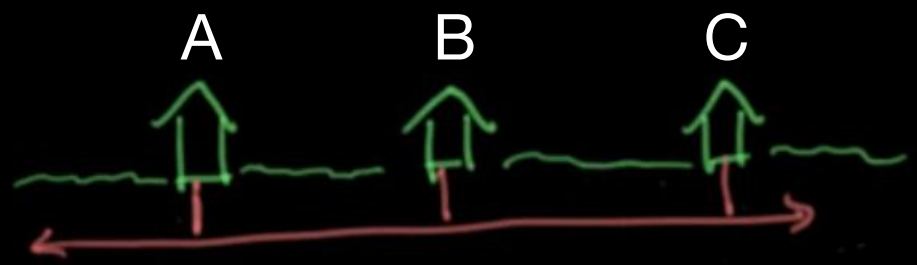


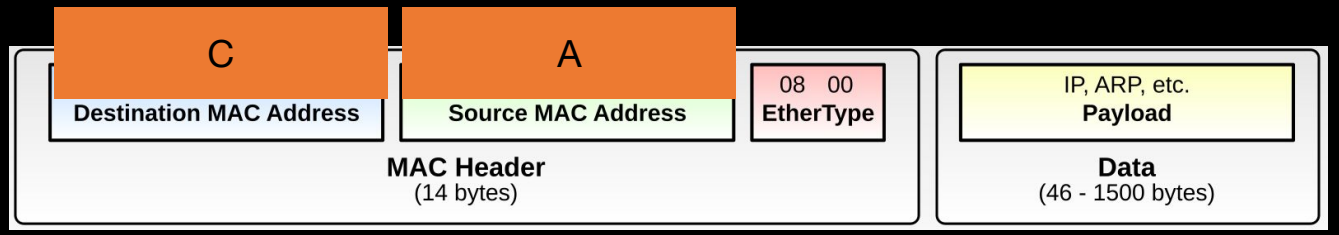


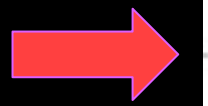


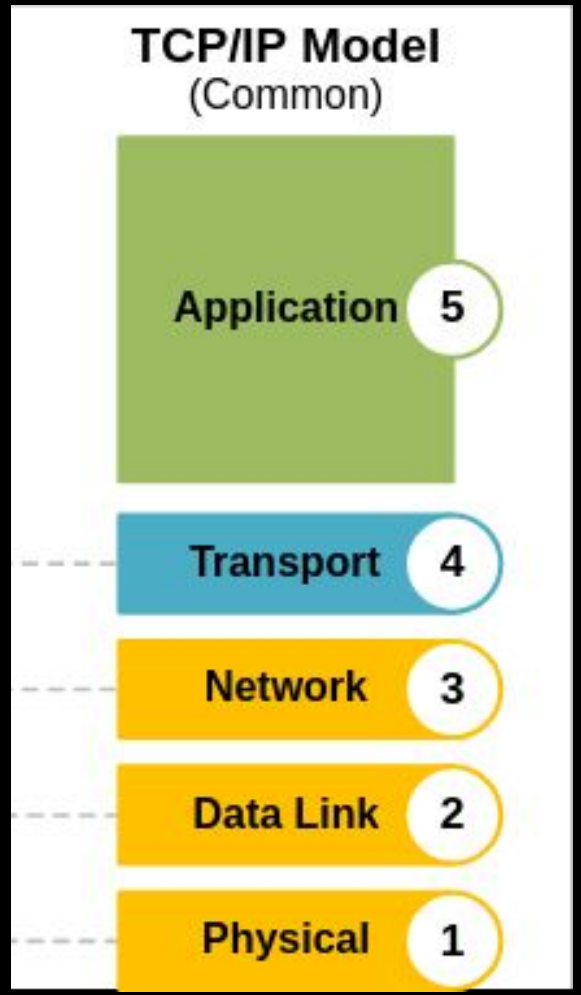
# Data Link Layer

- Give every device an address



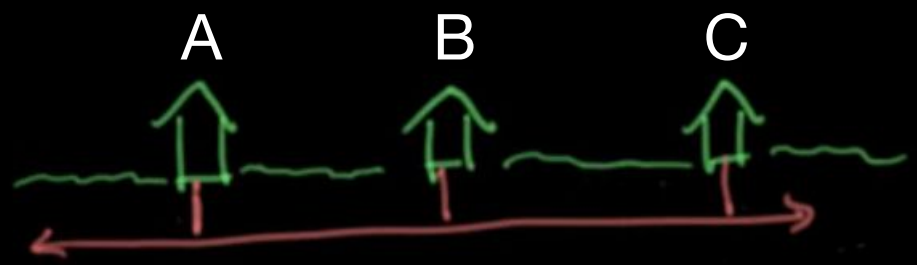



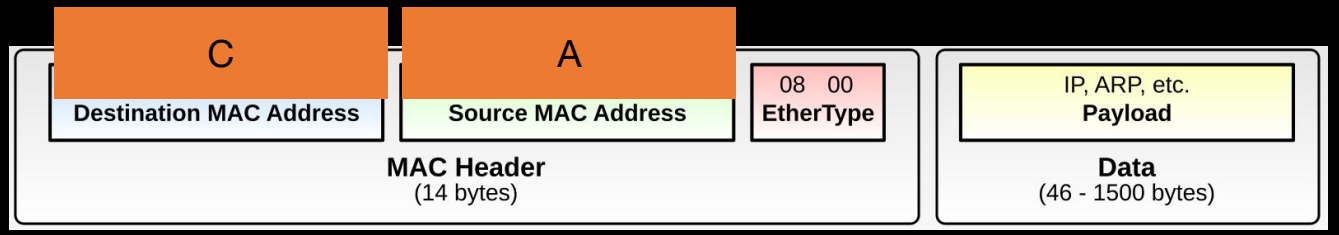




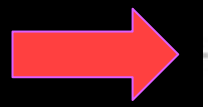
# Data Link Layer

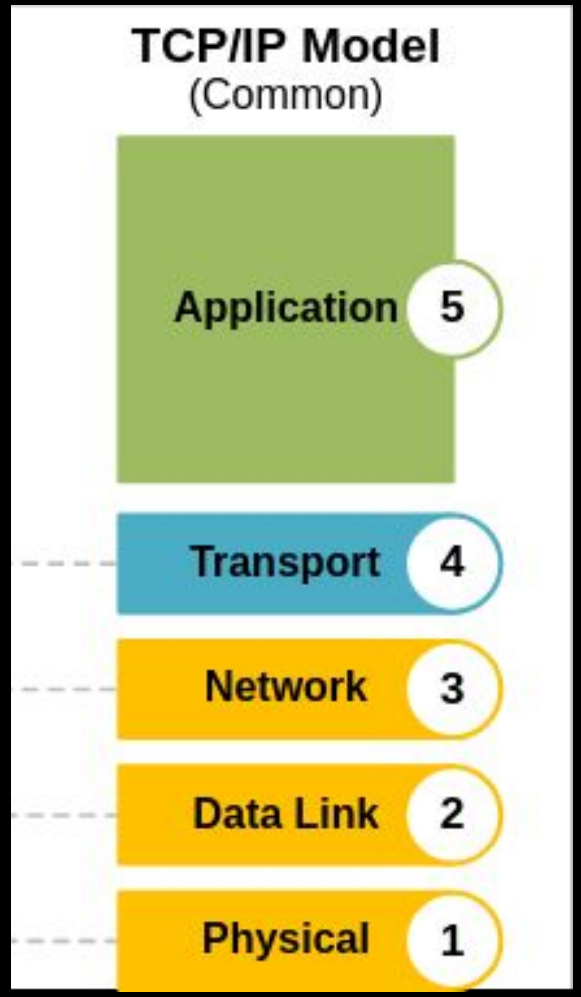
- Give every device an address





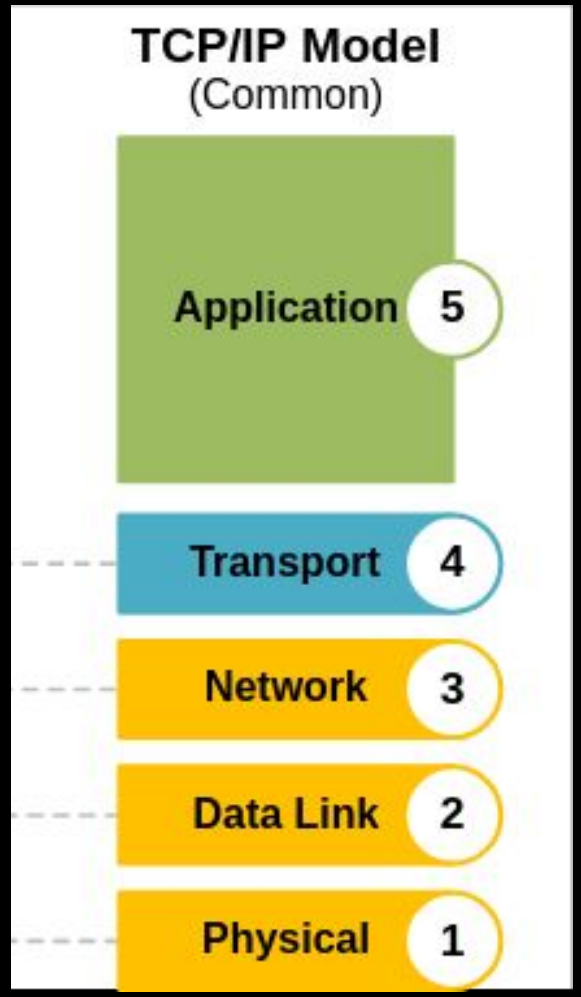





# Data Link Layer

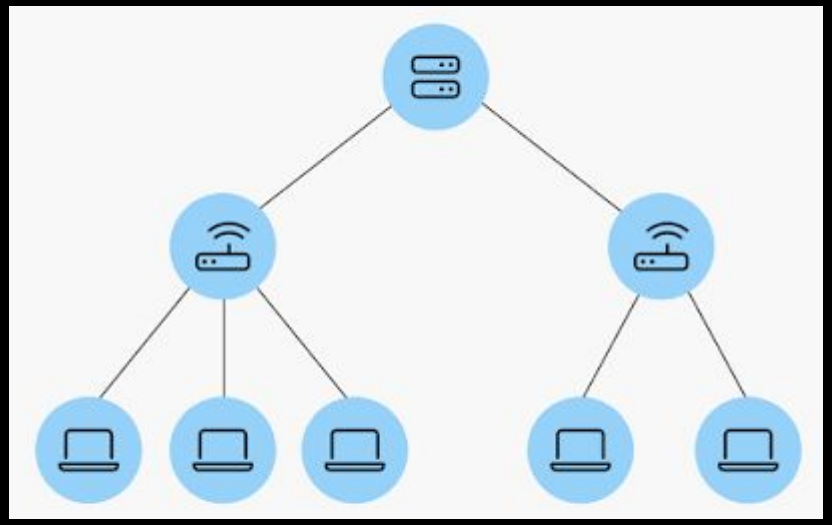
- On your laptop, run:
  - ip a (Linux)
  - ifconfig (Mac + some Linux)
  - ipconfig (Windows)
- Find your MAC address:
  looks like 6c:02:e0:6b:de:b4



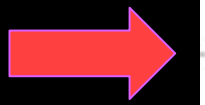


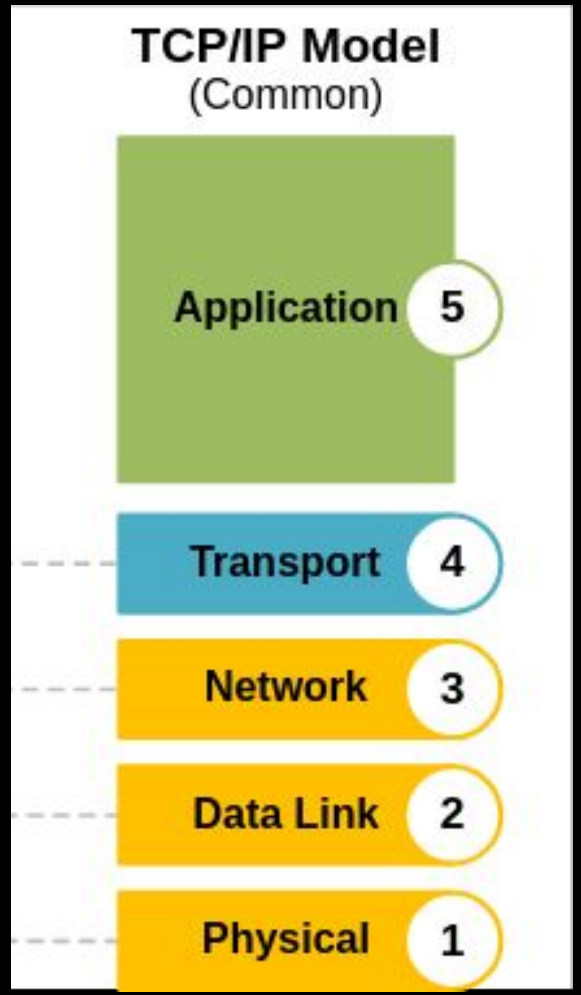
# Data Link Layer

- Give every device an address



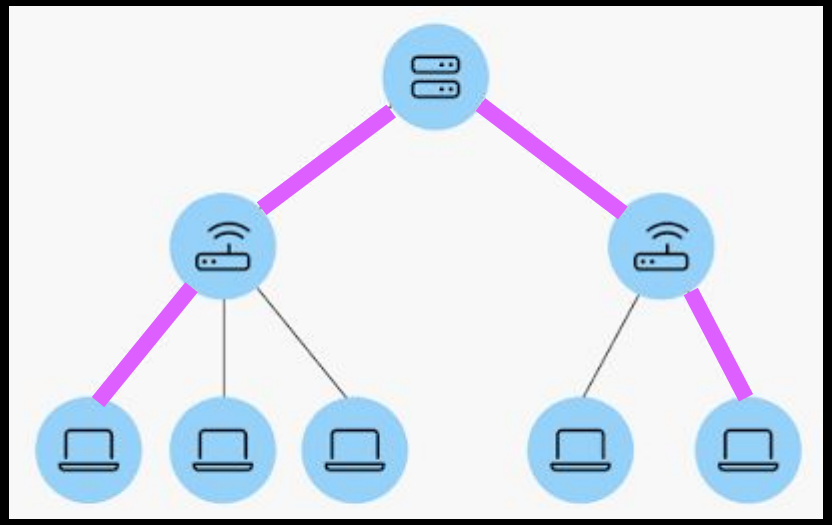
A B

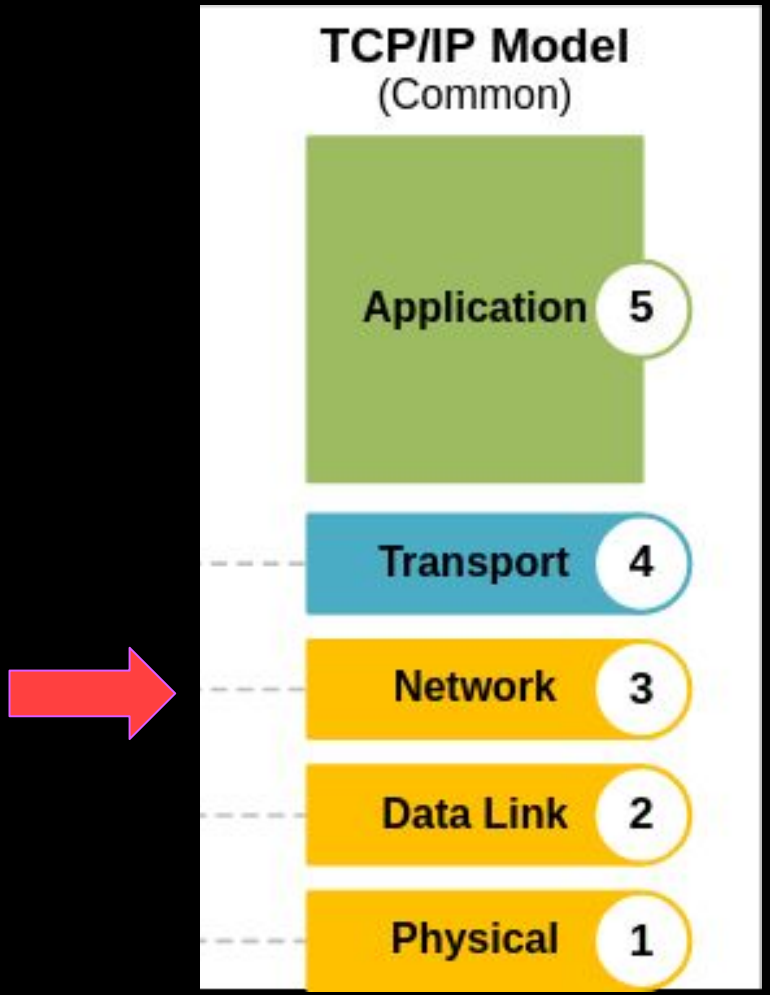



# Network Layer

- How do you route the data through an actual “network”

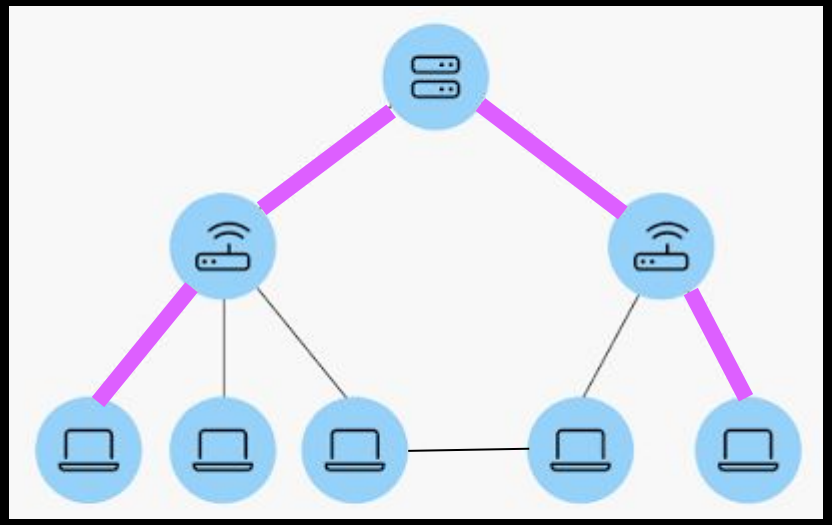


A B

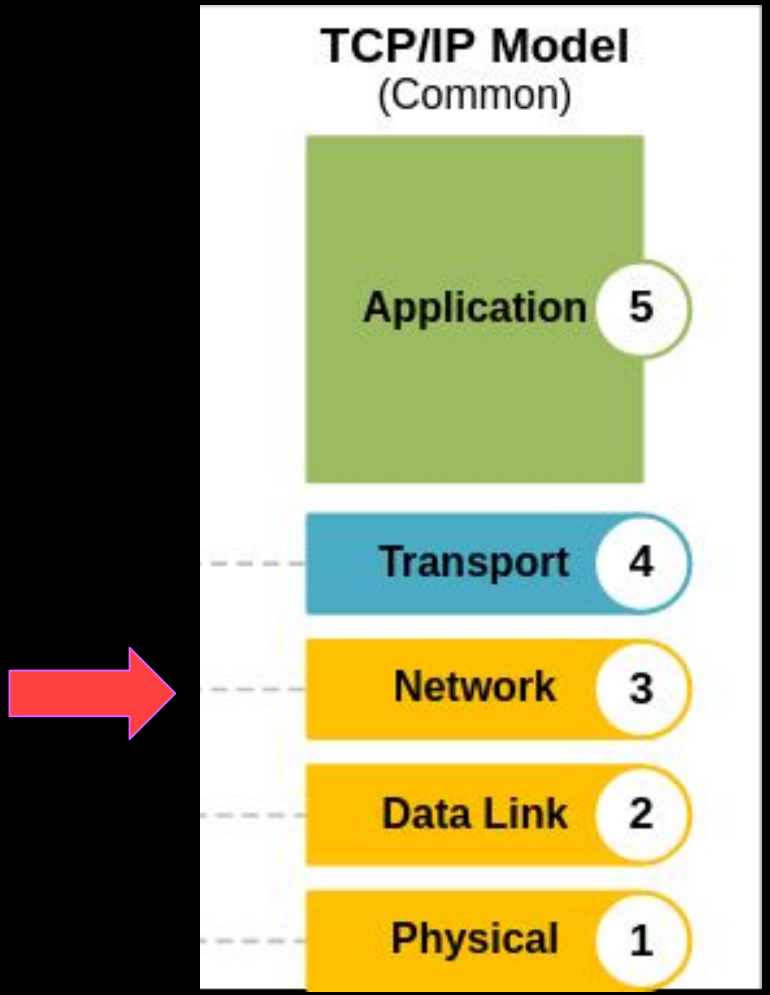



# Network Layer

- How do you route the data through an actual “network”

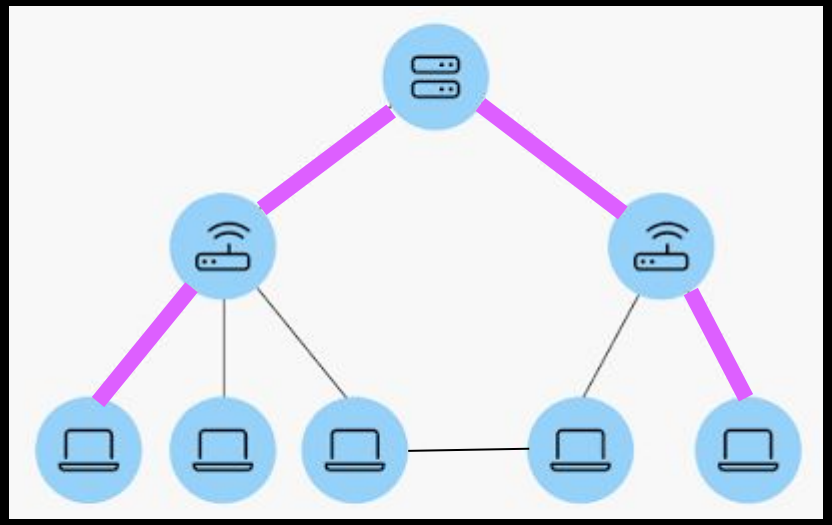


A B

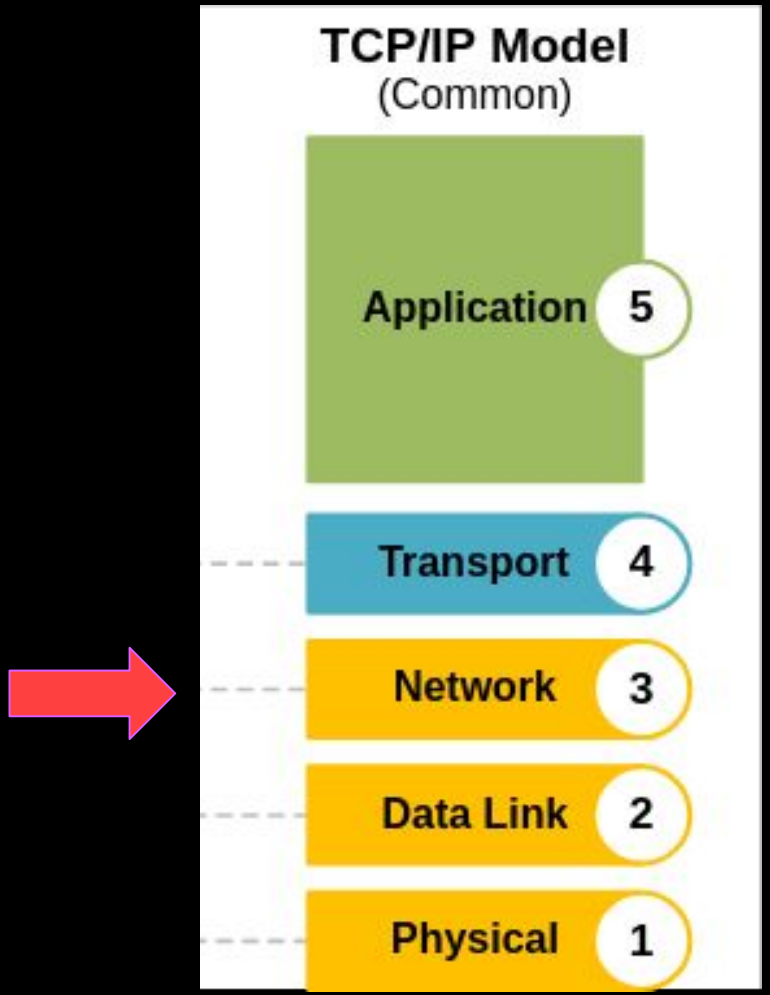



# Network Layer

- How do you route the data through an actual “network”

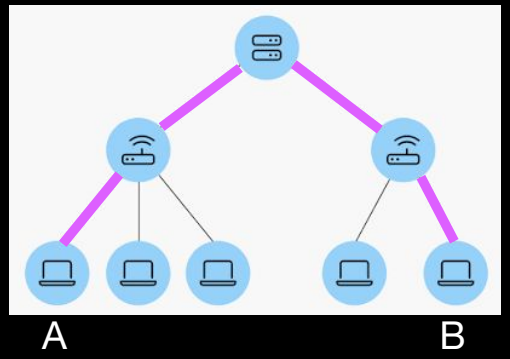


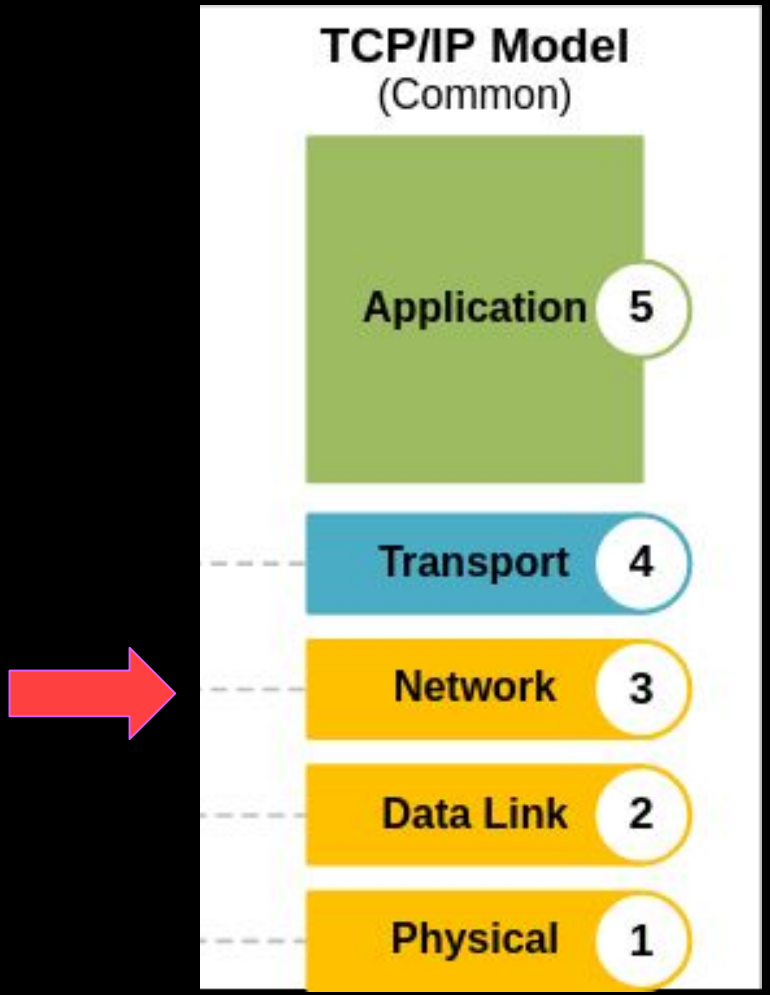
A B

# Network Layer

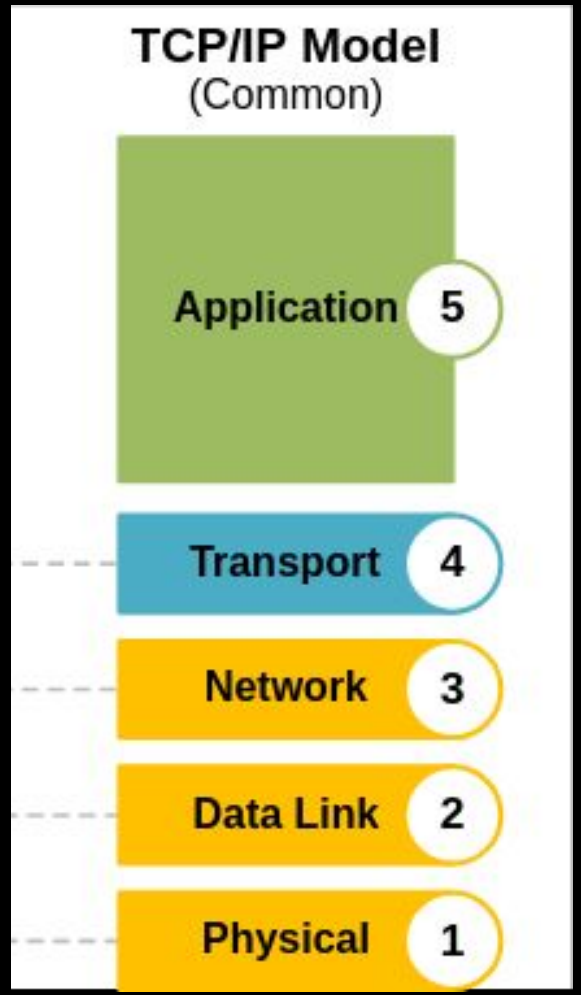
- On your laptop, run:
  - ip a (Linux)
  - ifconfig (Mac + some Linux)
  - ipconfig (Windows)
- Find your IP address:
  looks like 192.168.1.2
  or fe80::88c1:c4b3:1f93:7754
  or both!

# Now we abstract them away

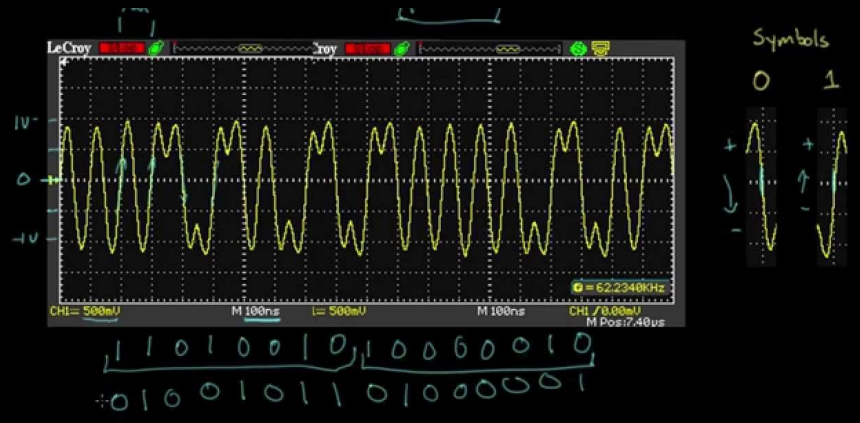
- In purple team, our focus is on the Transport and Application layers
- There are many ways to attack layers 1-3, but they are (mostly) out of scope
- We will touch on these attacks near the end

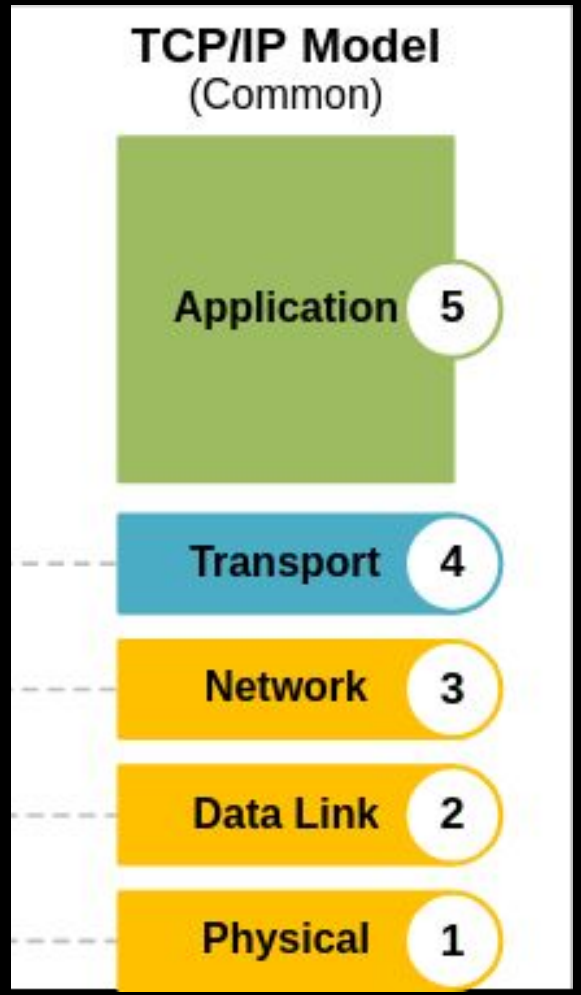




# If you want to learn more...

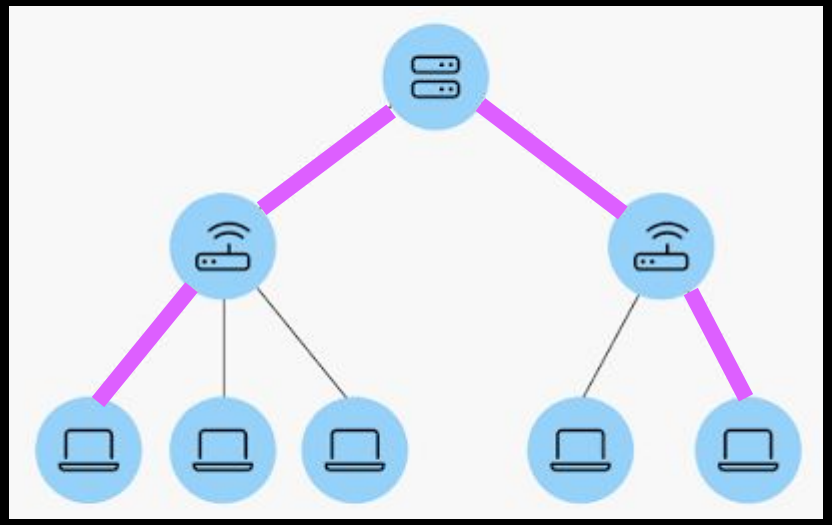
- In purple team, our focus is on the Transport and Application layers
- My favorite resource: [Ben Eater's Networking Tutorial]



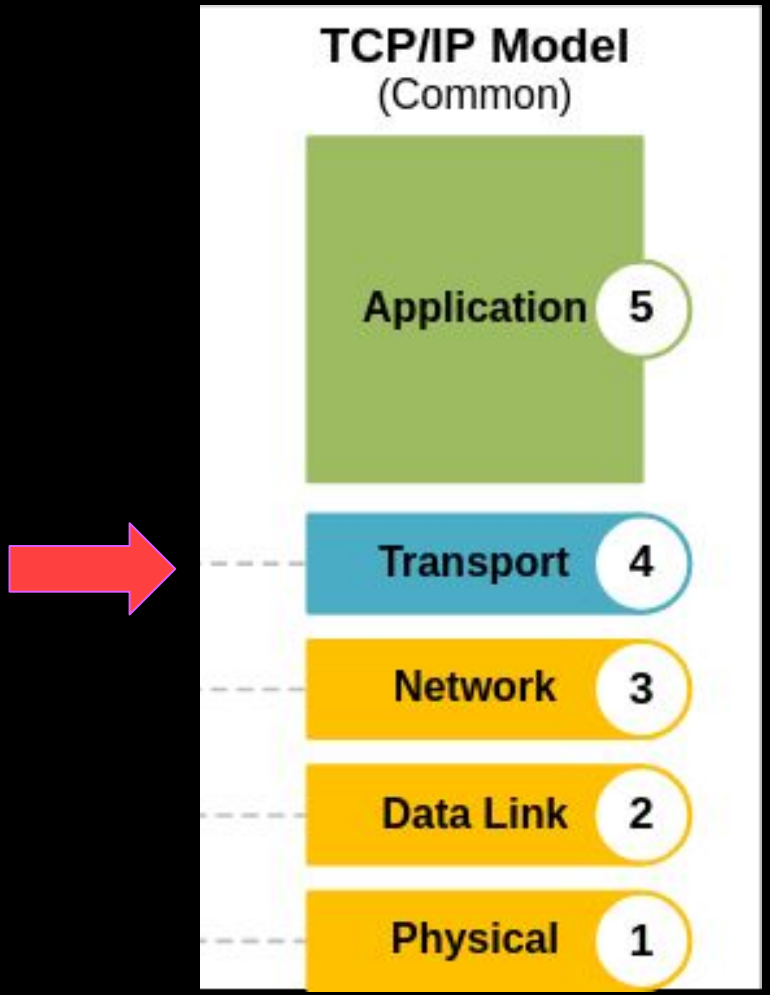



# Transport Layer

- Now that we know how to get there…

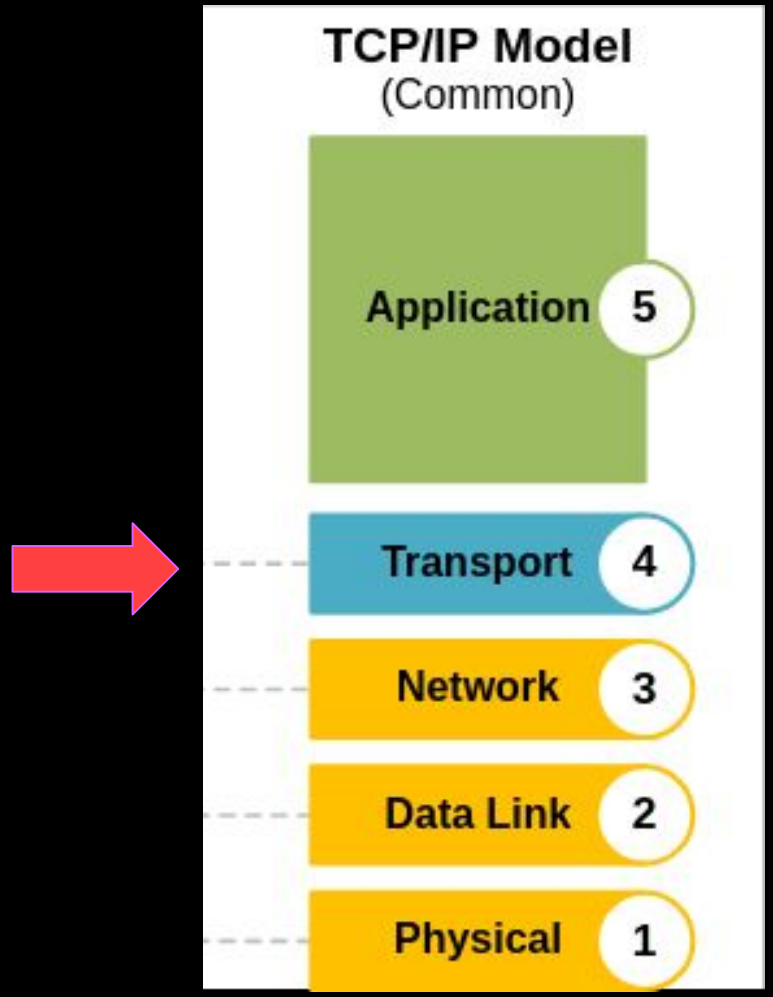


A B

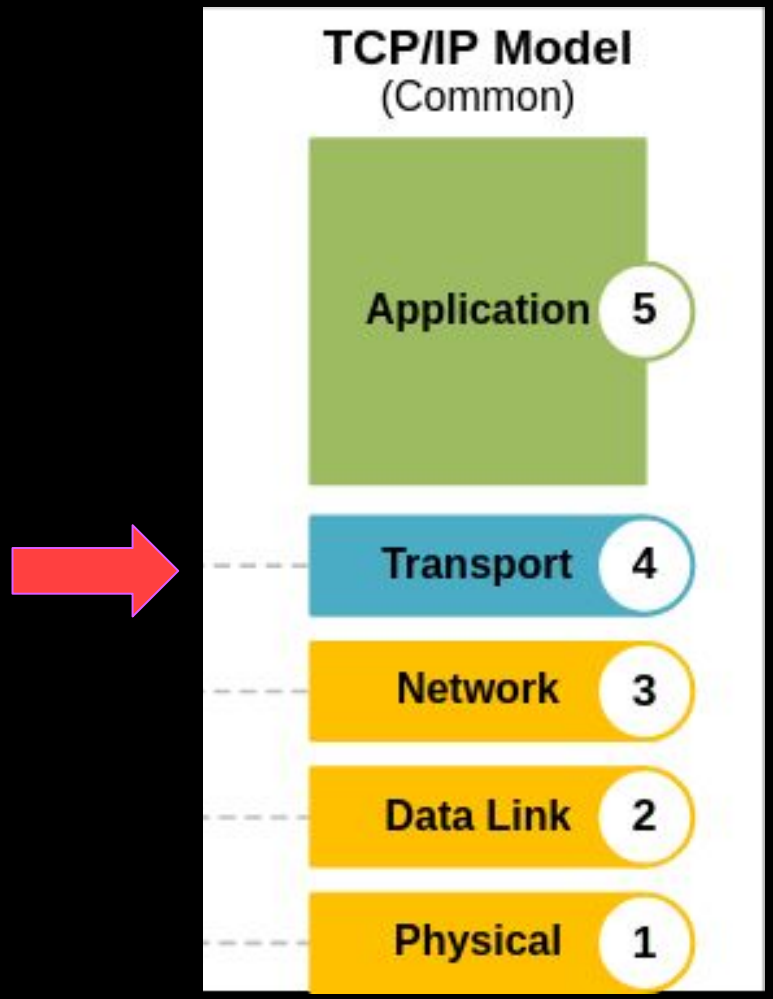
# Transport Layer

- Now that we know how to get there…
- How do we…
  - Split data into chunks?
  - Make sure they're in the right order?
  - Confirm that they were received?
  - Have multiple connections at the same time?
  - …and more

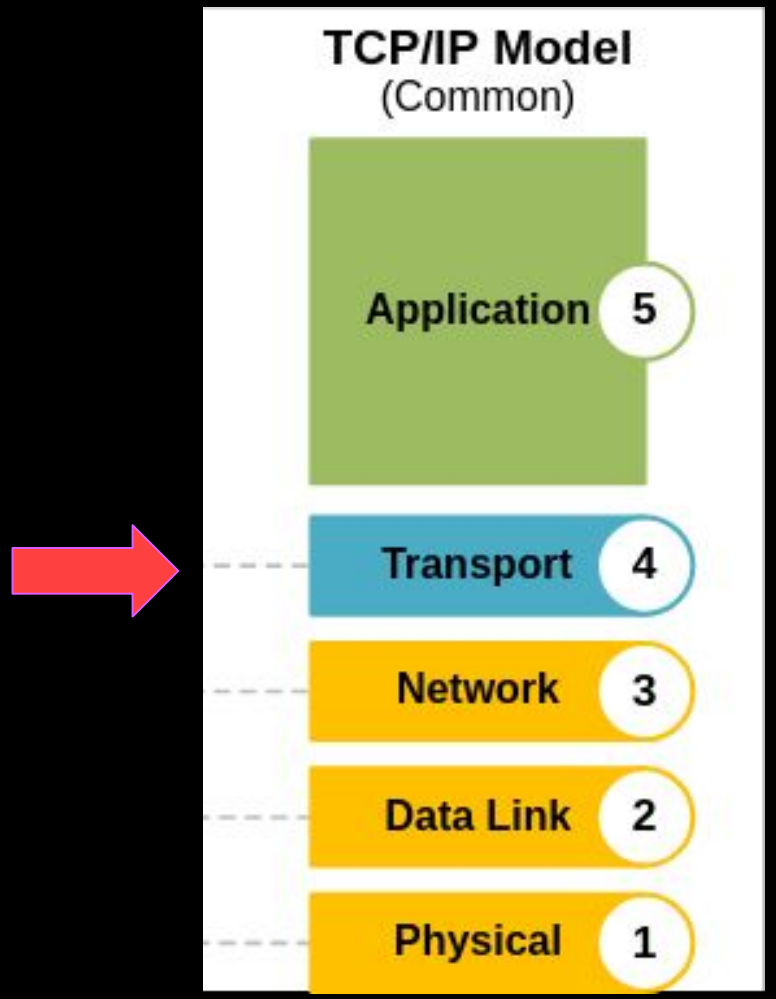

# Transport Layer

- Now that we know how to get there…
- How do we…
  - Split data into chunks?
  - Make sure they're in the right order?
  - Confirm that they were received?
  - Have multiple connections at the same time?
  - …and more

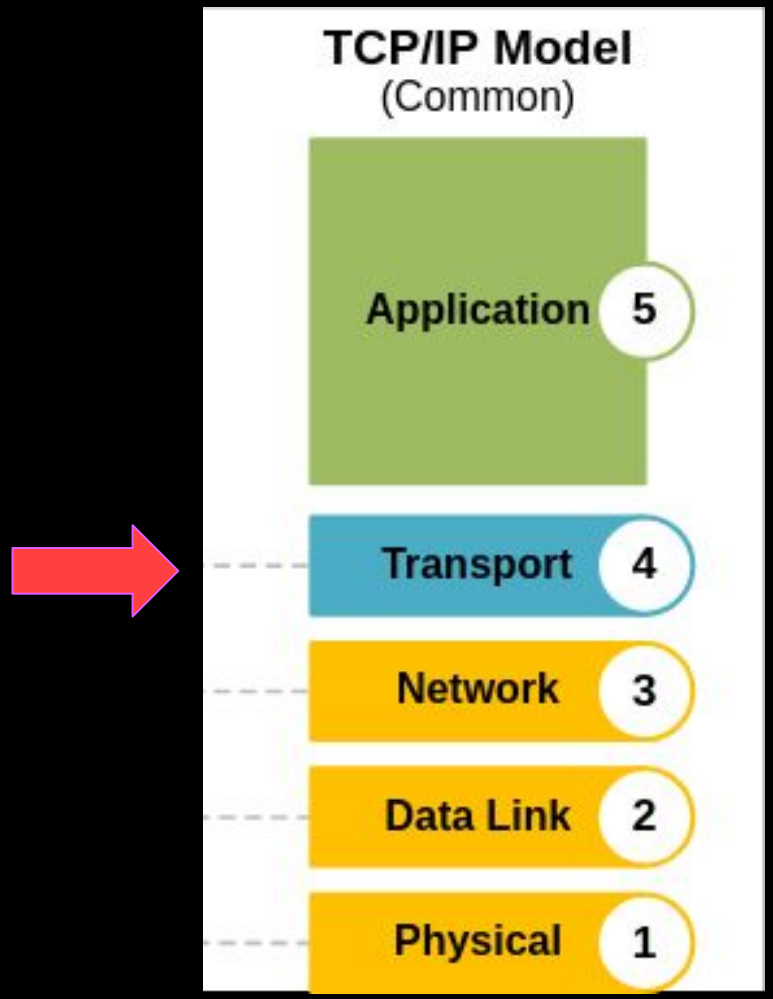
# Transport Layer

- Now that we know how to get there…
- How do we…
  - Split data into chunks?
  - Make sure they're in the right order?
  - Confirm that they were received?
  - Have multiple connections at the same time?
  - …and more

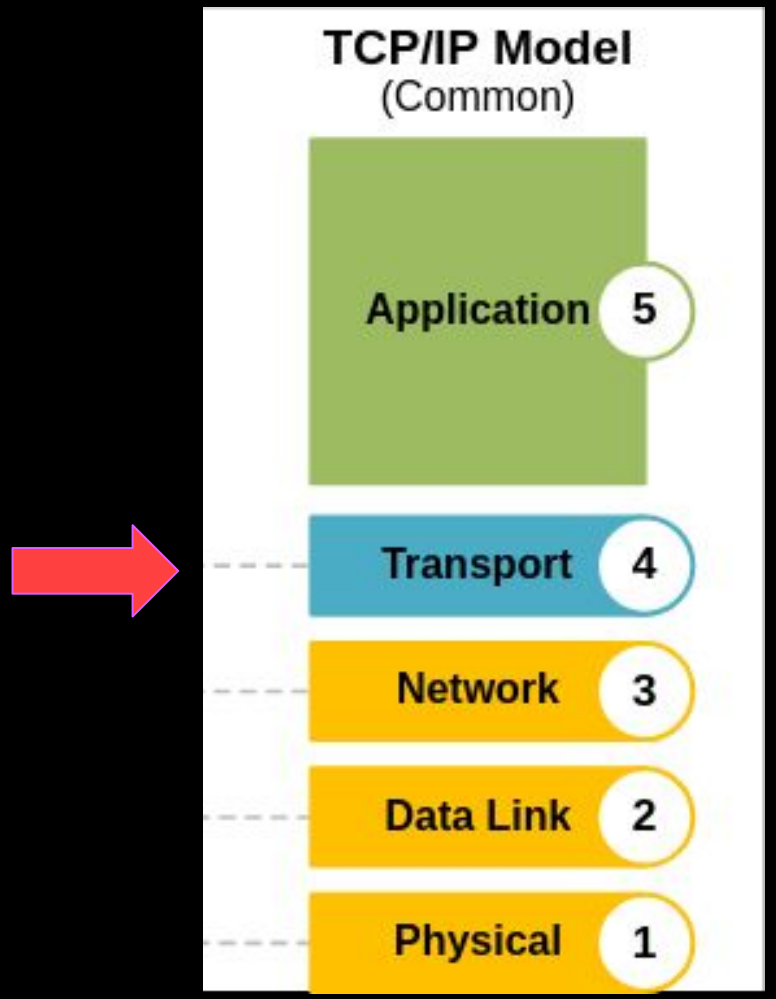

# Transport Layer

- How do we… Have multiple connections at the same time?



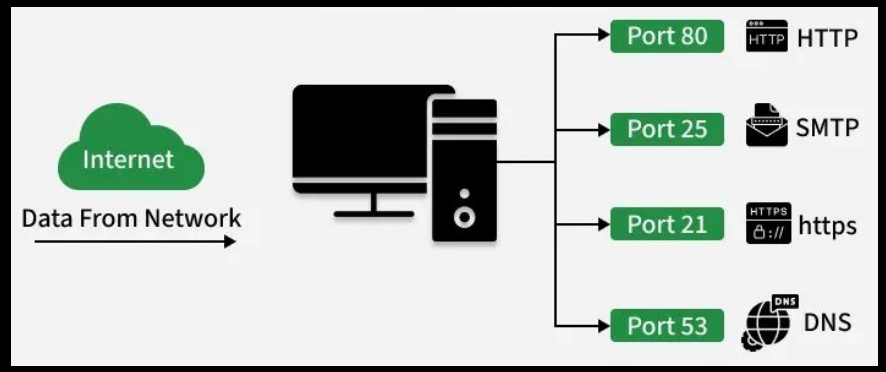

# Transport Layer

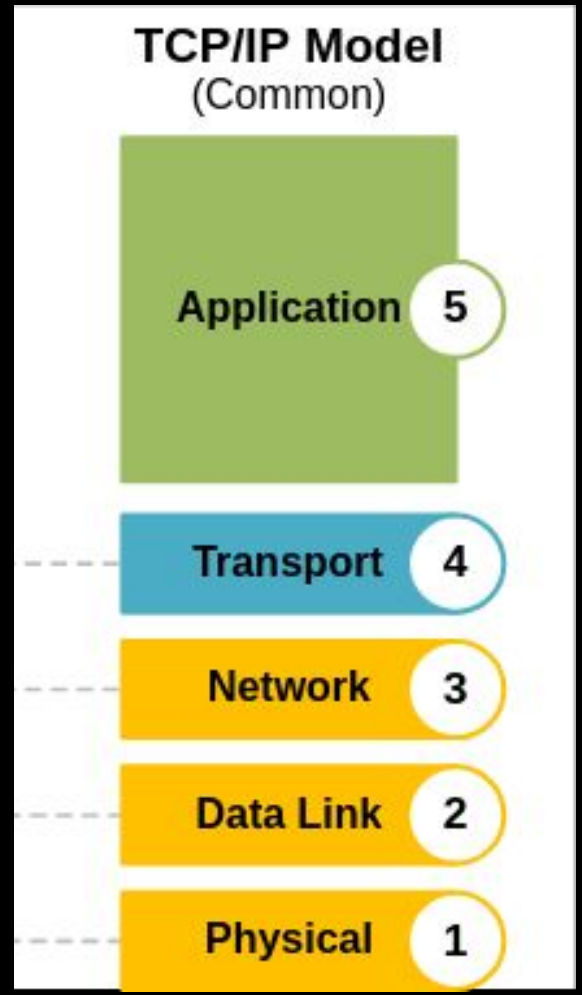
- How do we… Have multiple connections at the same time?
- Ports
  - Act as a logical endpoint for data
  - Each service/app can claim a port
  - Data meant for other services won't interfere

# Transport Layer

- How do we… Have multiple connections at the same time?
- Ports
  - Act as a logical endpoint for data
  - Each service/app can claim a port
  - Data meant for other services won't interfere

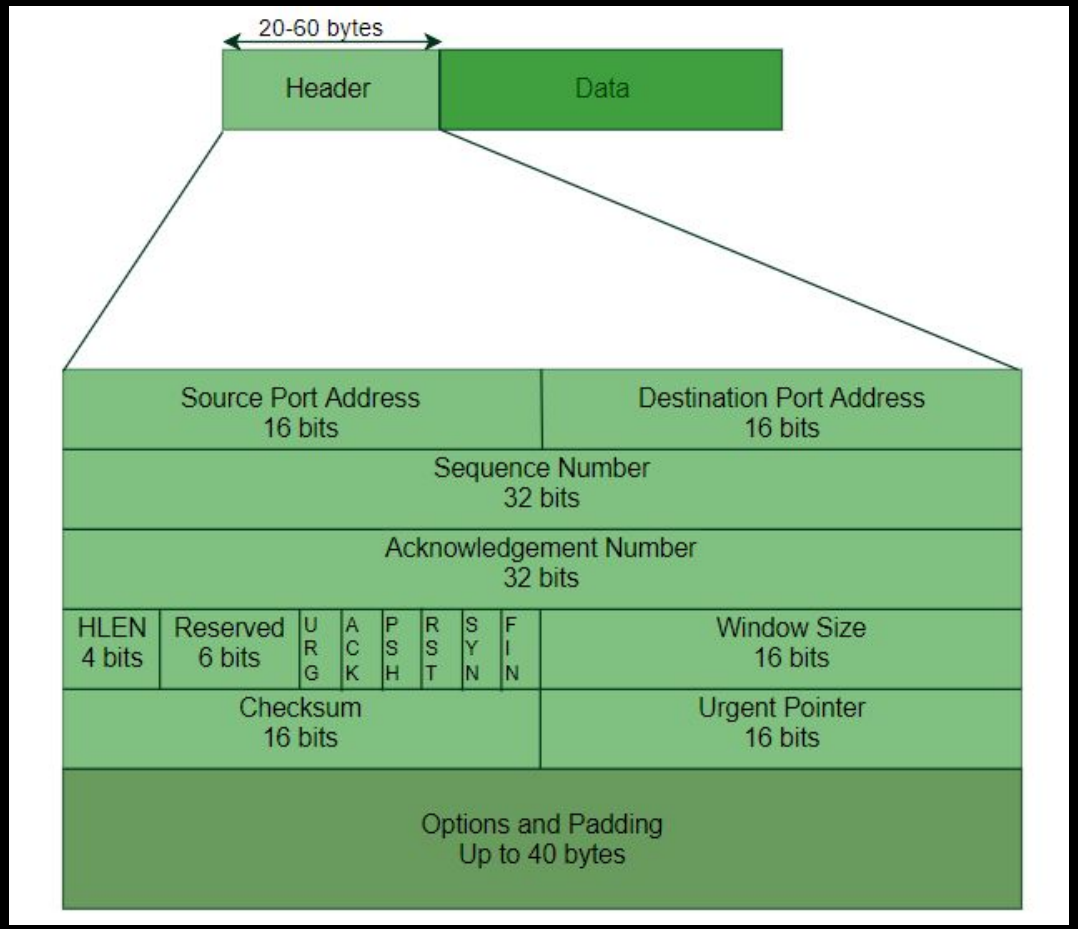



# Transport Layer

- How do we… Have multiple connections at the same time?
- Ports
  - Act as a logical endpoint for data
  - Each service/app can claim a port
  - Data meant for other services won't interfere

TCP Header





# TCP vs UDP

- TCP has a state machine to ensure reliability and speed.
- We will talk about the hand-shake process to initiate a connection, but during the connection it uses sequence number and acknowledgement number to make sure data is received reliably
- UDP is “best-effort”, data reliability is not guaranteed but has very low data overhead.
- Ideal for cases where speed matters over data integrity, like video streaming

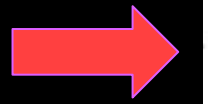
Most services you will see are going to be TCP!

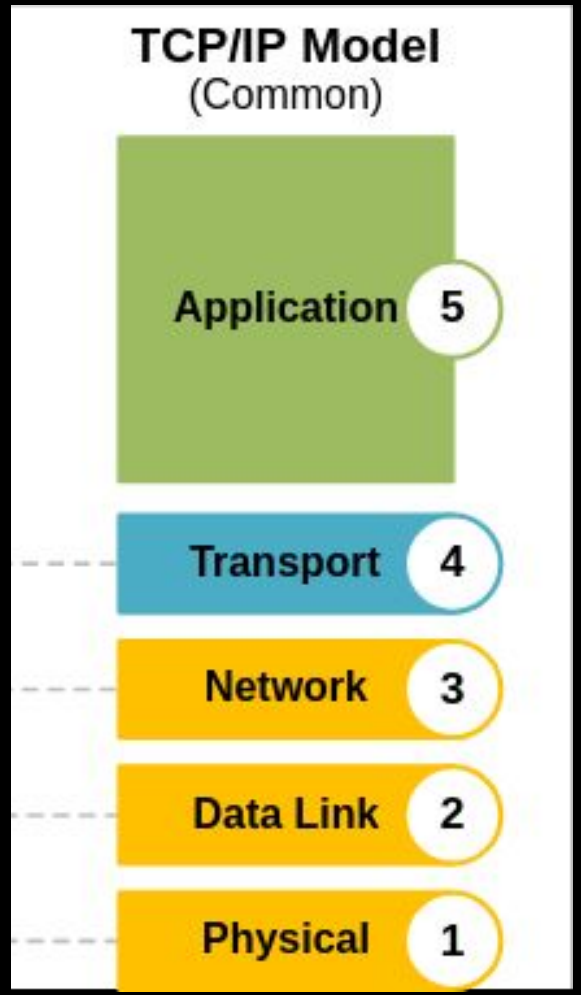
# Transmission Control Protocol

How do these sync?

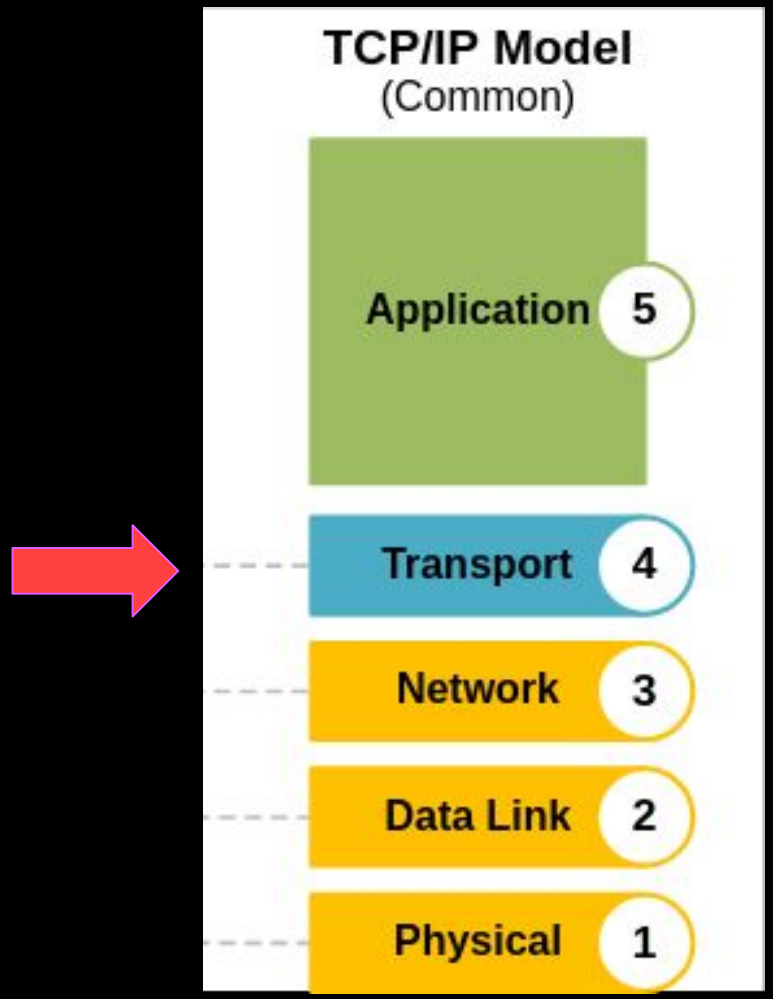
Client

Server

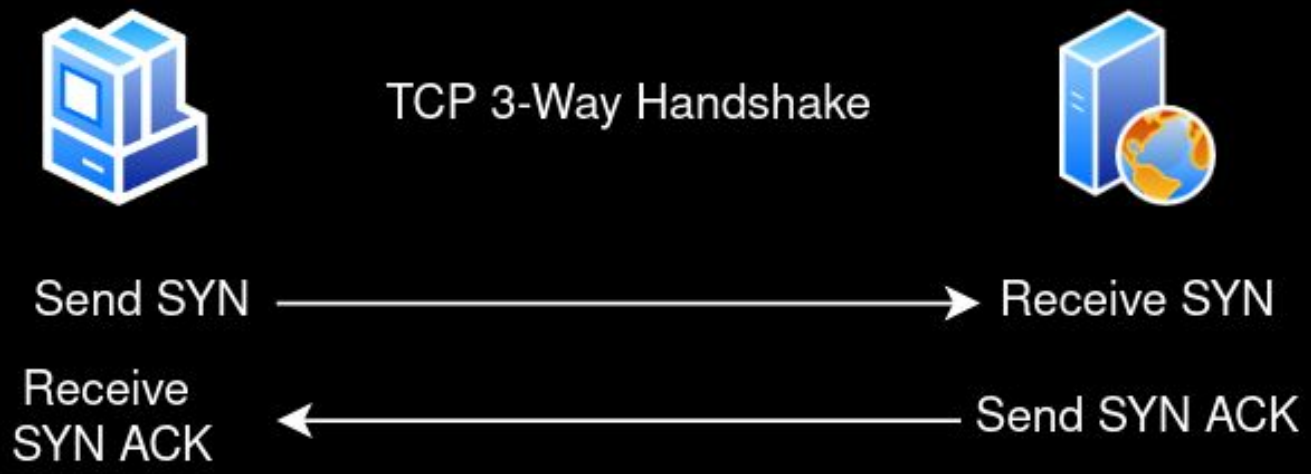


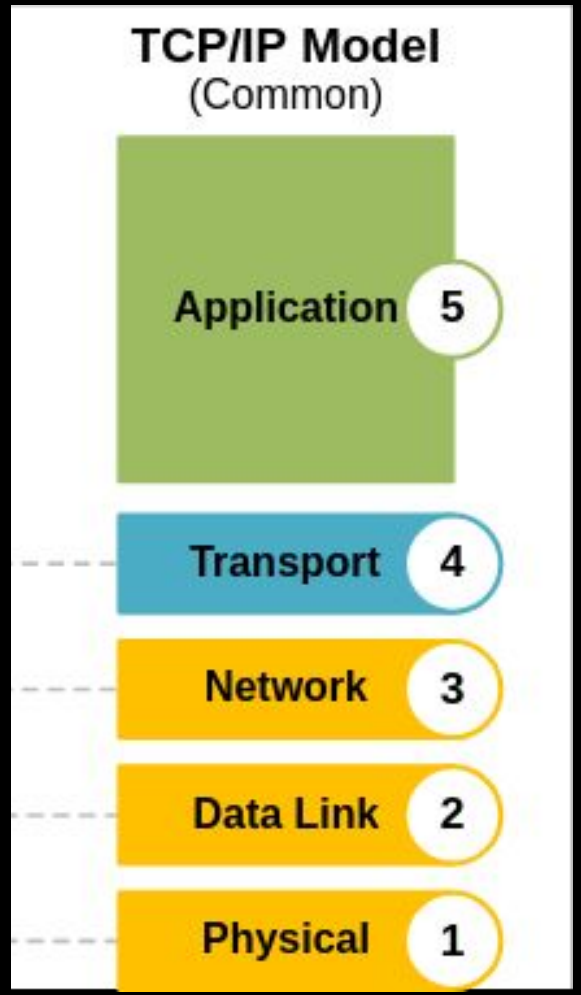


# Transmission Control Protocol

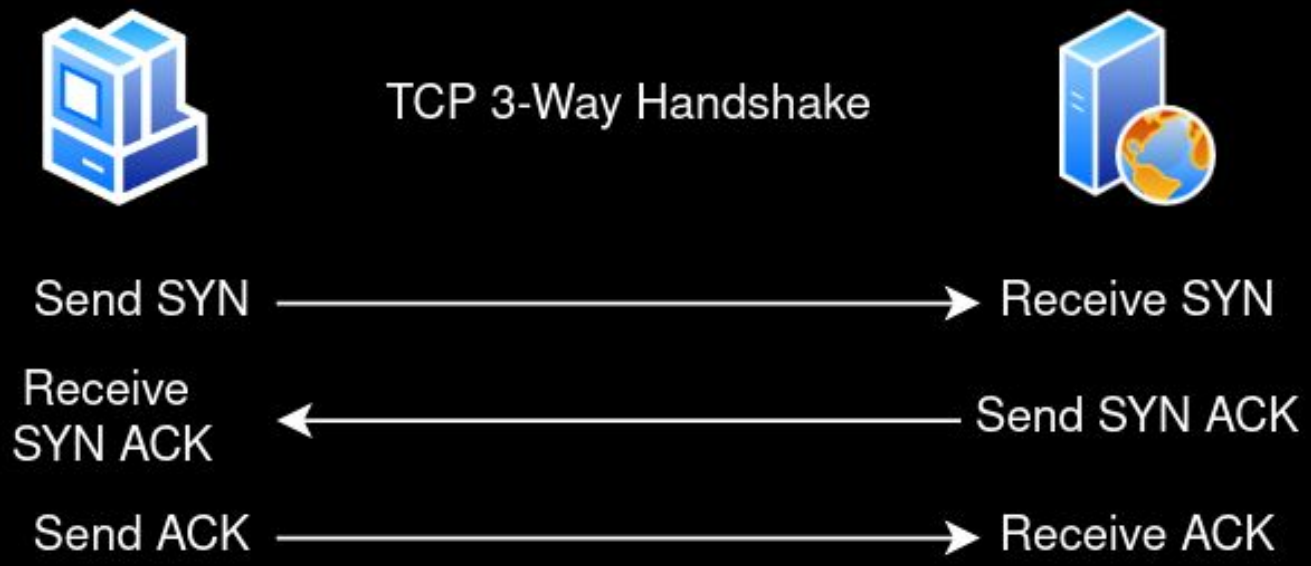



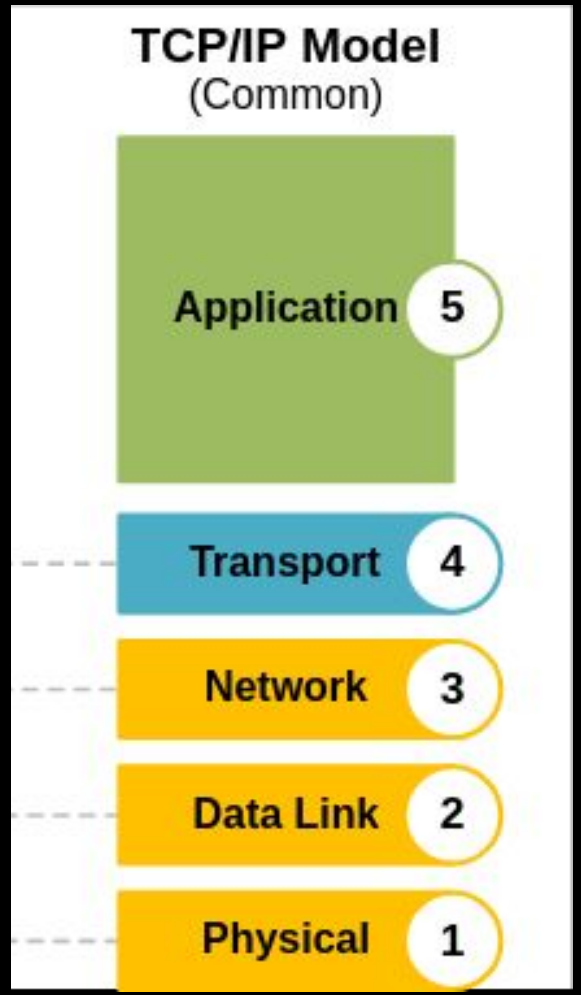
# Transmission Control Protocol



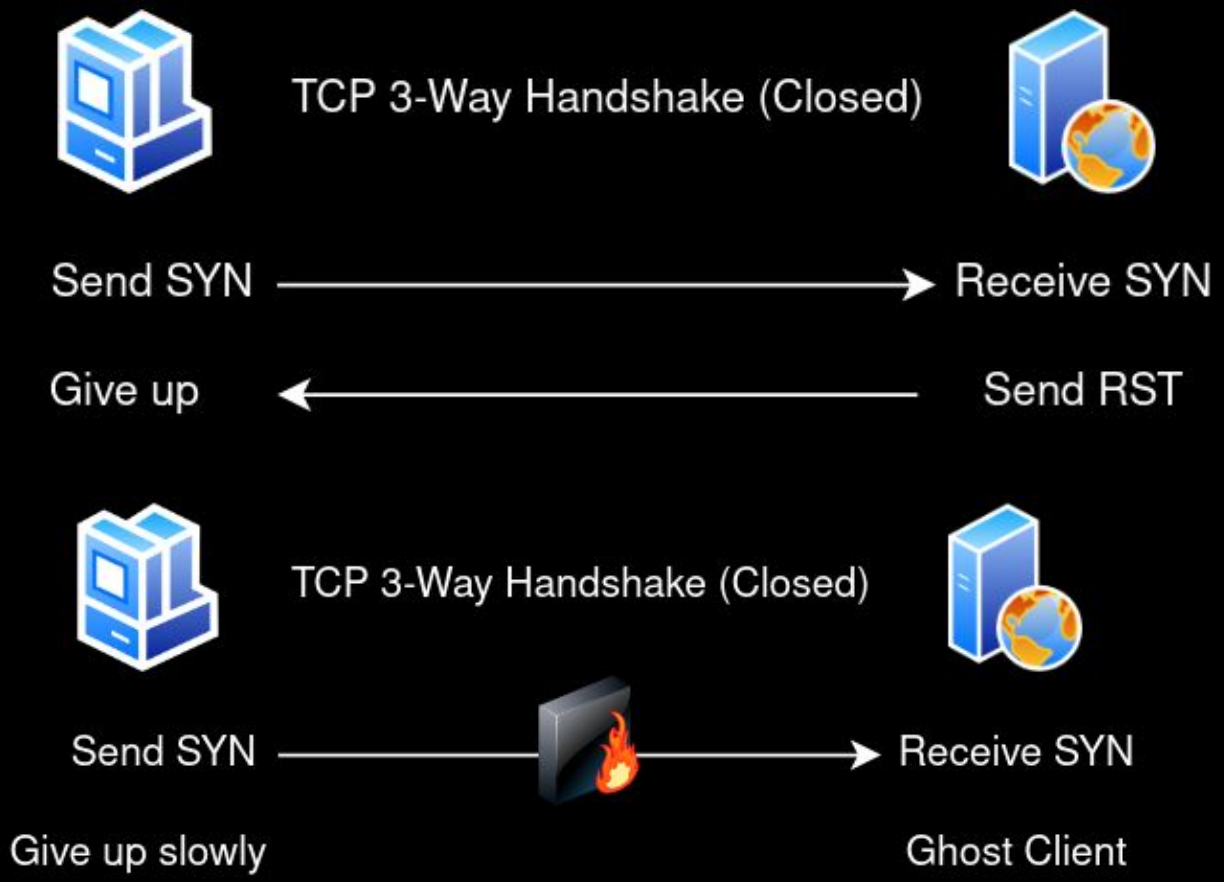



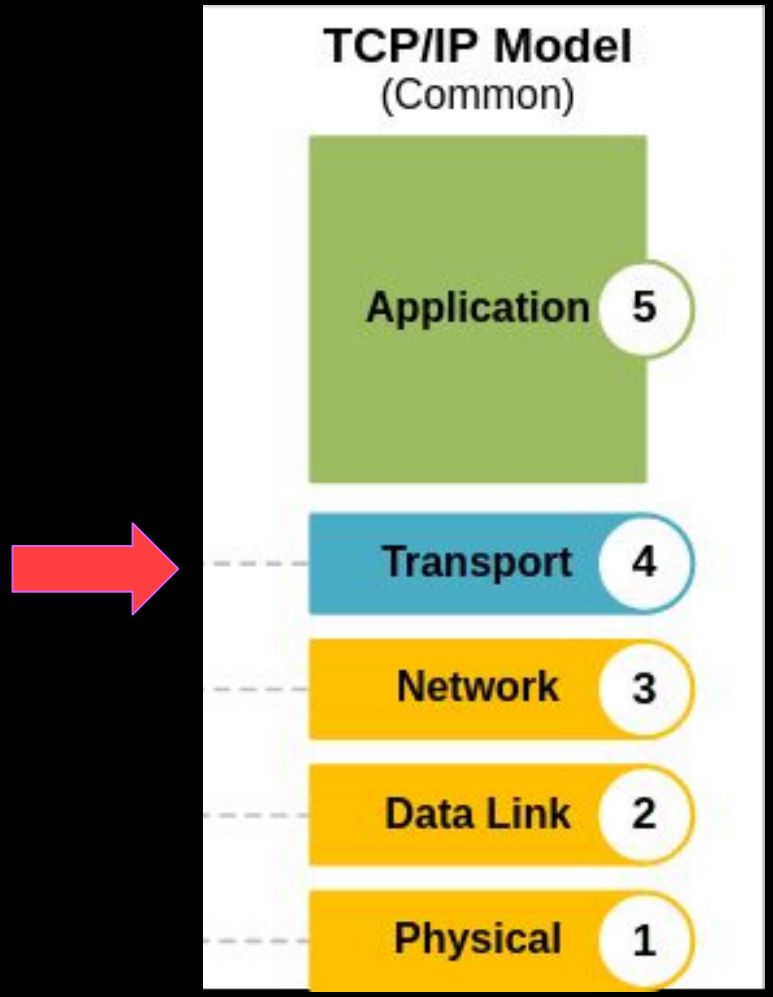
# Transmission Control Protocol



# Sorry, We're Closed



# Services

- Services serve content with specific Application protocols
- For example, an HTTP server is a service that serves web content with HTTP protocol


# Services

- Services serve content with specific Application protocols
- For example, an HTTP server is a service that serves web content with HTTP protocol


# Services

- Services serve content with specific Application protocols
- For example, an HTTP server is a service that serves web content with HTTP protocol

```
HTTP/1.1 200 OK
Content-Encoding: gzip
Age: 521648
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 06 Mar 2020 17:36:11 GMT
Etag: "3147526947+gzip"
Expires: Fri, 13 Mar 2020 17:36:11 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (dcb/7EC9)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 648
```


# Services

- Services serve content with specific Application protocols
- For example, an HTTP server is a service that serves web content with HTTP protocol
- Network Security concerns the security of **services** & **trust relationships** that occur in networked environments
- Common exploitable services include HTTP/S servers, SMB, SSH, NFS, **SQL, WinRM, and many more


# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)

# Services

- On your Kali VM, run either:
  - sudo ss -tunlp
  - sudo netstat -tunlp

```
~ ❯ sudo ss -tunlp
Netid            State                      Local Address:Port                      Peer Address:Port            Process
udp              UNCONN                           0.0.0.0:51003                          0.0.0.0:*                users:(("zen",pid=2865,fd=81))
udp              UNCONN                           0.0.0.0:34694                          0.0.0.0:*                users:(("zen",pid=2865,fd=147))
udp              UNCONN                           0.0.0.0:5353                           0.0.0.0:*                users:(("avahi-daemon",pid=902,fd=11))
udp              UNCONN                           0.0.0.0:5353                           0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=17))
udp              UNCONN                           0.0.0.0:5355                           0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=13))
udp              UNCONN                           0.0.0.0:38198                          0.0.0.0:*                users:(("zen",pid=2865,fd=70))
udp              UNCONN                           0.0.0.0:40931                          0.0.0.0:*                users:(("zen",pid=2865,fd=195))
udp              UNCONN                           0.0.0.0:41641                          0.0.0.0:*                users:((".tailscaled-wra",pid=1540,fd=19))
udp              UNCONN                           0.0.0.0:59410                          0.0.0.0:*
udp              UNCONN                           0.0.0.0:59602                          0.0.0.0:*                users:(("zen",pid=2865,fd=71))
udp              UNCONN                           0.0.0.0:43563                          0.0.0.0:*
udp              UNCONN                           0.0.0.0:44364                          0.0.0.0:*                users:(("zen",pid=2865,fd=250))
udp              UNCONN                           0.0.0.0:46409                          0.0.0.0:*                users:(("zen",pid=2865,fd=187))
udp              UNCONN                           0.0.0.0:47153                          0.0.0.0:*                users:(("zen",pid=2865,fd=197))
udp              UNCONN                        127.0.0.54:53                             0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=24))
udp              UNCONN                     127.0.0.53%lo:53                             0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=22))
udp              UNCONN                              *:1716                                   *:*                users:((".kdeconnectd-wr",pid=2384,fd=16))
udp              UNCONN                              *:53485                                  *:*                users:(("java",pid=592576,fd=368))
udp              UNCONN                         [::]:5353                                [::]:*                users:(("systemd-resolve",pid=547,fd=18))
udp              UNCONN                         [::]:5353                                [::]:*                users:(("avahi-daemon",pid=902,fd=12))
udp              UNCONN                         [::]:5355                                [::]:*                users:(("systemd-resolve",pid=547,fd=15))
udp              UNCONN                            *:56172                                  *:*                users:(("java",pid=592679,fd=307))
udp              UNCONN                         [::]:41641                               [::]:*                users:((".tailscaled-wra",pid=1540,fd=29))
udp              UNCONN                         [::]:59410                               [::]:*
udp              UNCONN                         [::]:43563                               [::]:*
tcp              LISTEN                     127.0.0.53%lo:53                             0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=23))
tcp              LISTEN                           0.0.0.0:5355                           0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=14))
tcp              LISTEN                       127.0.0.1:24891                            0.0.0.0:*                users:(("code",pid=591922,fd=65))
tcp              LISTEN                         0.0.0.0:22                               0.0.0.0:*                users:(("sshd",pid=1057,fd=6))
tcp              LISTEN                       127.0.0.1:6379                             0.0.0.0:*                users:(("redis-server",pid=1092,fd=9))
tcp              LISTEN                       127.0.0.1:6463                             0.0.0.0:*                users:(("electron",pid=280895,fd=131))
tcp              LISTEN                     100.95.242.59:55940                          0.0.0.0:*                users:((".tailscaled-wra",pid=1540,fd=33))
tcp              LISTEN                       127.0.0.1:40855                            0.0.0.0:*                users:(("code",pid=590648,fd=46))
tcp              LISTEN                       127.0.0.1:40739                            0.0.0.0:*                users:((".tailscaled-wra",pid=1712,fd=4))
tcp              LISTEN                       127.0.0.1:23635                            0.0.0.0:*                users:(("tinymist",pid=576458,fd=12))
tcp              LISTEN                       127.0.0.1:631                              0.0.0.0:*                users:(("cupsd",pid=1105,fd=4),("systemd",pid=1,fd=809))
tcp              LISTEN                    127.0.0.54:53                                 0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=25))
tcp              LISTEN                         [::]:5355                                [::]:*                users:(("systemd-resolve",pid=547,fd=16))
tcp              LISTEN                [::ffff:127.0.0.1]:46413                             *:*                users:(("java",pid=592679,fd=310))
tcp              LISTEN                      [::1]:631                                  [::]:*                users:(("cupsd",pid=1105,fd=8))
tcp              LISTEN                         [::]:22                                  [::]:*                users:(("sshd",pid=1057,fd=7))
tcp              LISTEN           [fd7a:115c:a1e0::a01:f23b]:55614                       [::]:*                users:((".tailscaled-wra",pid=1540,fd=34))
tcp              LISTEN                            *:1716                                   *:*                users:((".kdeconnectd-wr",pid=2384,fd=17))
```

```
~ ❯ sudo ss -tunlp
Netid            State                      Local Address:Port                     Peer Address:Port            Process
udp              UNCONN                           0.0.0.0:51003                         0.0.0.0:*              users:(("zen",pid=2865,fd=81))
udp              UNCONN                           0.0.0.0:34694                         0.0.0.0:*              users:(("zen",pid=2865,fd=147))
udp              UNCONN                           0.0.0.0:5353                          0.0.0.0:*              users:(("avahi-daemon",pid=902,fd=11))
udp              UNCONN                           0.0.0.0:5353                          0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=17))
udp              UNCONN                           0.0.0.0:5355                          0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=13))
udp              UNCONN                           0.0.0.0:38198                         0.0.0.0:*              users:(("zen",pid=2865,fd=70))
udp              UNCONN                           0.0.0.0:40931                         0.0.0.0:*              users:(("zen",pid=2865,fd=195))
udp              UNCONN                           0.0.0.0:41641                         0.0.0.0:*              users:((".tailscaled-wra",pid=1540,fd=19))
udp              UNCONN                           0.0.0.0:59410                         0.0.0.0:*
udp              UNCONN                           0.0.0.0:59602                         0.0.0.0:*              users:(("zen",pid=2865,fd=71))
udp              UNCONN                           0.0.0.0:43563                         0.0.0.0:*
udp              UNCONN                           0.0.0.0:44364                         0.0.0.0:*              users:(("zen",pid=2865,fd=250))
udp              UNCONN                           0.0.0.0:46409                         0.0.0.0:*              users:(("zen",pid=2865,fd=187))
udp              UNCONN                           0.0.0.0:47153                         0.0.0.0:*              users:(("zen",pid=2865,fd=197))
udp              UNCONN                        127.0.0.54:53                            0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=24))
udp              UNCONN                     127.0.0.53%lo:53                            0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=22))
udp              UNCONN                              *:1716                                  *:*              users:((".kdeconnectd-wr",pid=2384,fd=16))
udp              UNCONN                              *:53485                                 *:*              users:(("java",pid=592576,fd=368))
udp              UNCONN                         [::]:5353                               [::]:*              users:(("systemd-resolve",pid=547,fd=18))
udp              UNCONN                         [::]:5353                               [::]:*              users:(("avahi-daemon",pid=902,fd=12))
udp              UNCONN                         [::]:5355                               [::]:*              users:(("systemd-resolve",pid=547,fd=15))
udp              UNCONN                            *:56172                                 *:*              users:(("java",pid=592679,fd=307))
udp              UNCONN                         [::]:41641                              [::]:*              users:((".tailscaled-wra",pid=1540,fd=29))
udp              UNCONN                         [::]:59410                              [::]:*
udp              UNCONN                         [::]:43563                              [::]:*
tcp              LISTEN                     127.0.0.53%lo:53                            0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=23))
tcp              LISTEN                           0.0.0.0:5355                          0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=14))
tcp              LISTEN                       127.0.0.1:24891                           0.0.0.0:*              users:(("code",pid=591922,fd=65))
tcp              LISTEN                           0.0.0.0:22                            0.0.0.0:*              users:(("sshd",pid=1057,fd=6))
tcp              LISTEN                       127.0.0.1:6379                            0.0.0.0:*              users:(("redis-server",pid=1092,fd=9))
tcp              LISTEN                       127.0.0.1:6463                            0.0.0.0:*              users:(("electron",pid=280895,fd=131))
tcp              LISTEN                   100.95.242.59:55940                           0.0.0.0:*              users:((".tailscaled-wra",pid=1540,fd=33))
tcp              LISTEN                       127.0.0.1:40855                           0.0.0.0:*              users:(("code",pid=590648,fd=46))
tcp              LISTEN                       127.0.0.1:40739                           0.0.0.0:*              users:((".tailscaled-wra",pid=1712,fd=4))
tcp              LISTEN                       127.0.0.1:23635                           0.0.0.0:*              users:(("tinymist",pid=576458,fd=12))
tcp              LISTEN                       127.0.0.1:631                             0.0.0.0:*              users:(("cupsd",pid=1105,fd=4),("systemd",pid=1,fd=809))
tcp              LISTEN                    127.0.0.54:53                                0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=25))
tcp              LISTEN                         [::]:5355                               [::]:*              users:(("systemd-resolve",pid=547,fd=16))
tcp              LISTEN                [::ffff:127.0.0.1]:46413                            *:*              users:(("java",pid=592679,fd=310))
tcp              LISTEN                      [::1]:631                                [::]:*              users:(("cupsd",pid=1105,fd=8))
tcp              LISTEN                         [::]:22                                 [::]:*              users:(("sshd",pid=1057,fd=7))
tcp              LISTEN          [fd7a:115c:a1e0::a01:f23b]:55614                       [::]:*              users:((".tailscaled-wra",pid=1540,fd=34))
tcp              LISTEN                            *:1716                                  *:*              users:((".kdeconnectd-wr",pid=2384,fd=17))
```

```
~ ❯ sudo ss -tunlp
Netid            State                      Local Address:Port                     Peer Address:Port            Process
udp              UNCONN                           0.0.0.0:51003                         0.0.0.0:*              users:(("zen",pid=2865,fd=81))
udp              UNCONN                           0.0.0.0:34694                         0.0.0.0:*              users:(("zen",pid=2865,fd=147))
udp              UNCONN                           0.0.0.0:5353                          0.0.0.0:*              users:(("avahi-daemon",pid=902,fd=11))
udp              UNCONN                           0.0.0.0:5353                          0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=17))
udp              UNCONN                           0.0.0.0:5355                          0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=13))
udp              UNCONN                           0.0.0.0:38198                         0.0.0.0:*              users:(("zen",pid=2865,fd=70))
udp              UNCONN                           0.0.0.0:40931                         0.0.0.0:*              users:(("zen",pid=2865,fd=195))
udp              UNCONN                           0.0.0.0:41641                         0.0.0.0:*              users:((".tailscaled-wra",pid=1540,fd=19))
udp              UNCONN                           0.0.0.0:59410                         0.0.0.0:*
udp              UNCONN                           0.0.0.0:59602                         0.0.0.0:*              users:(("zen",pid=2865,fd=71))
udp              UNCONN                           0.0.0.0:43563                         0.0.0.0:*
udp              UNCONN                           0.0.0.0:44364                         0.0.0.0:*              users:(("zen",pid=2865,fd=250))
udp              UNCONN                           0.0.0.0:46409                         0.0.0.0:*              users:(("zen",pid=2865,fd=187))
udp              UNCONN                           0.0.0.0:47153                         0.0.0.0:*              users:(("zen",pid=2865,fd=197))
udp              UNCONN                        127.0.0.54:53                            0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=24))
udp              UNCONN                     127.0.0.53%lo:53                            0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=22))
udp              UNCONN                              *:1716                                  *:*              users:((".kdeconnectd-wr",pid=2384,fd=16))
udp              UNCONN                              *:53485                                 *:*              users:(("java",pid=592576,fd=368))
udp              UNCONN                         [::]:5353                               [::]:*              users:(("systemd-resolve",pid=547,fd=18))
udp              UNCONN                         [::]:5353                               [::]:*              users:(("avahi-daemon",pid=902,fd=12))
udp              UNCONN                         [::]:5355                               [::]:*              users:(("systemd-resolve",pid=547,fd=15))
udp              UNCONN                            *:56172                                 *:*              users:(("java",pid=592679,fd=307))
udp              UNCONN                         [::]:41641                              [::]:*              users:((".tailscaled-wra",pid=1540,fd=29))
udp              UNCONN                         [::]:59410                              [::]:*
udp              UNCONN                         [::]:43563                              [::]:*
tcp              LISTEN                     127.0.0.53%lo:53                            0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=23))
tcp              LISTEN                           0.0.0.0:5355                          0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=14))
tcp              LISTEN                       127.0.0.1:24891                           0.0.0.0:*              users:(("code",pid=591922,fd=65))
tcp              LISTEN                           0.0.0.0:22                            0.0.0.0:*              users:(("sshd",pid=1057,fd=6))
tcp              LISTEN                       127.0.0.1:6379                            0.0.0.0:*              users:(("redis-server",pid=1092,fd=9))
tcp              LISTEN                       127.0.0.1:6463                            0.0.0.0:*              users:(("electron",pid=280895,fd=131))
tcp              LISTEN                   100.95.242.59:55940                           0.0.0.0:*              users:((".tailscaled-wra",pid=1540,fd=33))
tcp              LISTEN                       127.0.0.1:40855                           0.0.0.0:*              users:(("code",pid=590648,fd=46))
tcp              LISTEN                       127.0.0.1:40739                           0.0.0.0:*              users:((".tailscaled-wra",pid=1712,fd=4))
tcp              LISTEN                       127.0.0.1:23635                           0.0.0.0:*              users:(("tinymist",pid=576458,fd=12))
tcp              LISTEN                       127.0.0.1:631                             0.0.0.0:*              users:(("cupsd",pid=1105,fd=4),("systemd",pid=1,fd=809))
tcp              LISTEN                    127.0.0.54:53                                0.0.0.0:*              users:(("systemd-resolve",pid=547,fd=25))
tcp              LISTEN                         [::]:5355                               [::]:*              users:(("systemd-resolve",pid=547,fd=16))
tcp              LISTEN                [::ffff:127.0.0.1]:46413                             *:*              users:(("java",pid=592679,fd=310))
tcp              LISTEN                      [::1]:631                                 [::]:*              users:(("cupsd",pid=1105,fd=8))
tcp              LISTEN                         [::]:22                                 [::]:*              users:(("sshd",pid=1057,fd=7))
tcp              LISTEN          [fd7a:115c:a1e0::a01:f23b]:55614                       [::]:*              users:((".tailscaled-wra",pid=1540,fd=34))
tcp              LISTEN                            *:1716                                  *:*              users:((".kdeconnectd-wr",pid=2384,fd=17))
```

```
~ ❯ sudo ss -tunlp
Netid            State                      Local Address:Port                      Peer Address:Port            Process
udp              UNCONN                           0.0.0.0:51003                          0.0.0.0:*                users:(("zen",pid=2865,fd=81))
udp              UNCONN                           0.0.0.0:34694                          0.0.0.0:*                users:(("zen",pid=2865,fd=147))
udp              UNCONN                           0.0.0.0:5353                           0.0.0.0:*                users:(("avahi-daemon",pid=902,fd=11))
udp              UNCONN                           0.0.0.0:5353                           0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=17))
udp              UNCONN                           0.0.0.0:5355                           0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=13))
udp              UNCONN                           0.0.0.0:38198                          0.0.0.0:*                users:(("zen",pid=2865,fd=70))
udp              UNCONN                           0.0.0.0:40931                          0.0.0.0:*                users:(("zen",pid=2865,fd=195))
udp              UNCONN                           0.0.0.0:41641                          0.0.0.0:*                users:((".tailscaled-wra",pid=1540,fd=19))
udp              UNCONN                           0.0.0.0:59410                          0.0.0.0:*
udp              UNCONN                           0.0.0.0:59602                          0.0.0.0:*                users:(("zen",pid=2865,fd=71))
udp              UNCONN                           0.0.0.0:43563                          0.0.0.0:*
udp              UNCONN                           0.0.0.0:44364                          0.0.0.0:*                users:(("zen",pid=2865,fd=250))
udp              UNCONN                           0.0.0.0:46409                          0.0.0.0:*                users:(("zen",pid=2865,fd=187))
udp              UNCONN                           0.0.0.0:47153                          0.0.0.0:*                users:(("zen",pid=2865,fd=197))
udp              UNCONN                        127.0.0.54:53                             0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=24))
udp              UNCONN                     127.0.0.53%lo:53                             0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=22))
udp              UNCONN                           *:1716                                    *:*                users:((".kdeconnectd-wr",pid=2384,fd=16))
udp              UNCONN                           *:53485                                   *:*                users:(("java",pid=592576,fd=368))
udp              UNCONN                      [::]:5353                                 [::]:*                users:(("systemd-resolve",pid=547,fd=18))
udp              UNCONN                      [::]:5353                                 [::]:*                users:(("avahi-daemon",pid=902,fd=12))
udp              UNCONN                      [::]:5355                                 [::]:*                users:(("systemd-resolve",pid=547,fd=15))
udp              UNCONN                         *:56172                                   *:*                users:(("java",pid=592679,fd=307))
udp              UNCONN                      [::]:41641                                [::]:*                users:((".tailscaled-wra",pid=1540,fd=29))
udp              UNCONN                      [::]:59410                                [::]:*
udp              UNCONN                      [::]:43563                                [::]:*
tcp              LISTEN                     127.0.0.53%lo:53                             0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=23))
tcp              LISTEN                           0.0.0.0:5355                           0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=14))
tcp              LISTEN                       127.0.0.1:24891                            0.0.0.0:*                users:(("code",pid=591922,fd=65))
tcp              LISTEN                           0.0.0.0:22                             0.0.0.0:*                users:(("sshd",pid=1057,fd=6))
tcp              LISTEN                       127.0.0.1:6379                             0.0.0.0:*                users:(("redis-server",pid=1092,fd=9))
tcp              LISTEN                       127.0.0.1:6463                             0.0.0.0:*                users:(("electron",pid=280895,fd=131))
tcp              LISTEN                     100.95.242.59:55940                          0.0.0.0:*                users:((".tailscaled-wra",pid=1540,fd=33))
tcp              LISTEN                       127.0.0.1:40855                            0.0.0.0:*                users:(("code",pid=590648,fd=46))
tcp              LISTEN                       127.0.0.1:40739                            0.0.0.0:*                users:((".tailscaled-wra",pid=1712,fd=4))
tcp              LISTEN                       127.0.0.1:23635                            0.0.0.0:*                users:(("tinymist",pid=576458,fd=12))
tcp              LISTEN                       127.0.0.1:631                              0.0.0.0:*                users:(("cupsd",pid=1105,fd=4),("systemd",pid=1,fd=809))
tcp              LISTEN                    127.0.0.54:53                                 0.0.0.0:*                users:(("systemd-resolve",pid=547,fd=25))
tcp              LISTEN                          [::]:5355                                 [::]:*                users:(("systemd-resolve",pid=547,fd=16))
tcp              LISTEN                [::ffff:127.0.0.1]:46413                             *:*                users:(("java",pid=592679,fd=310))
tcp              LISTEN                       [::1]:631                                 [::]:*                users:(("cupsd",pid=1105,fd=8))
tcp              LISTEN                          [::]:22                                   [::]:*                users:(("sshd",pid=1057,fd=7))
tcp              LISTEN          [fd7a:115c:a1e0::a01:f23b]:55614                          [::]:*                users:((".tailscaled-wra",pid=1540,fd=34))
tcp              LISTEN                           *:1716                                    *:*                users:((".kdeconnectd-wr",pid=2384,fd=17))
```

# Summary

- Data link layer allows communication between connected machines
- Internet layer offers a way to address and find machines (IPv4/v6)
- Transport layer offers end-to-end communications (TCP and UDP), with 65536 different ports each to run different connections on
- Application layer offers application-specific (e.g. client-server) communication without worrying about underlying implementation
- To fully understand a service, we must know the address, TCP/UDP and port, as well as application layer protocol
- e.g. https://sigpwny.com -> 172.66.x.x, TCP, port 443, HTTPS


# Port Scanning

# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)

# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)

All the ports above are the default ports, assigned by IANA!

# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)

```
~ ❯ ssh 192.168.1.1
ssh: connect to host 192.168.1.1 port 22: Connection refused
```

All the ports above are the default ports, assigned by IANA!

# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)
- **Non-Default HTTP** (31337)

# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)
- **Non-Default HTTP** (31337)

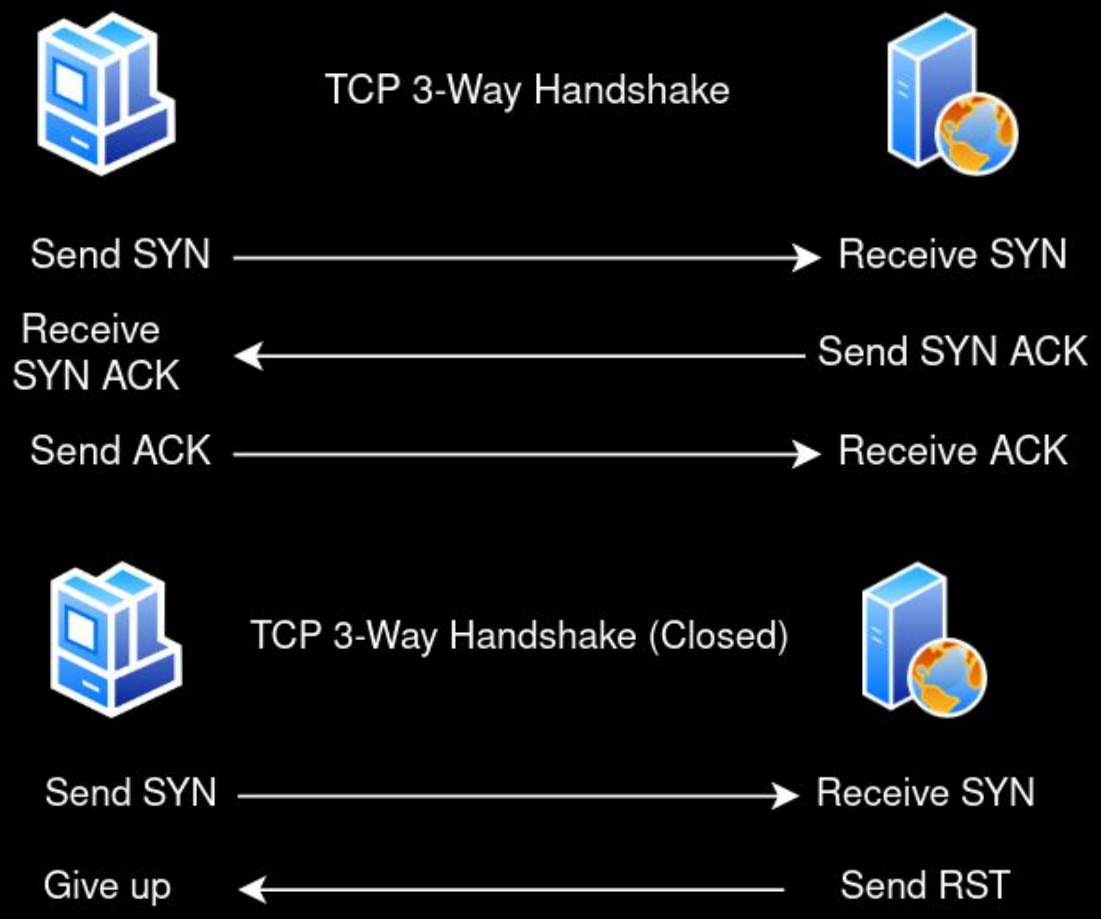
How do we tell **which services** are running on **which ports** for a given IP?

# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)
- **Non-Default HTTP** (31337)

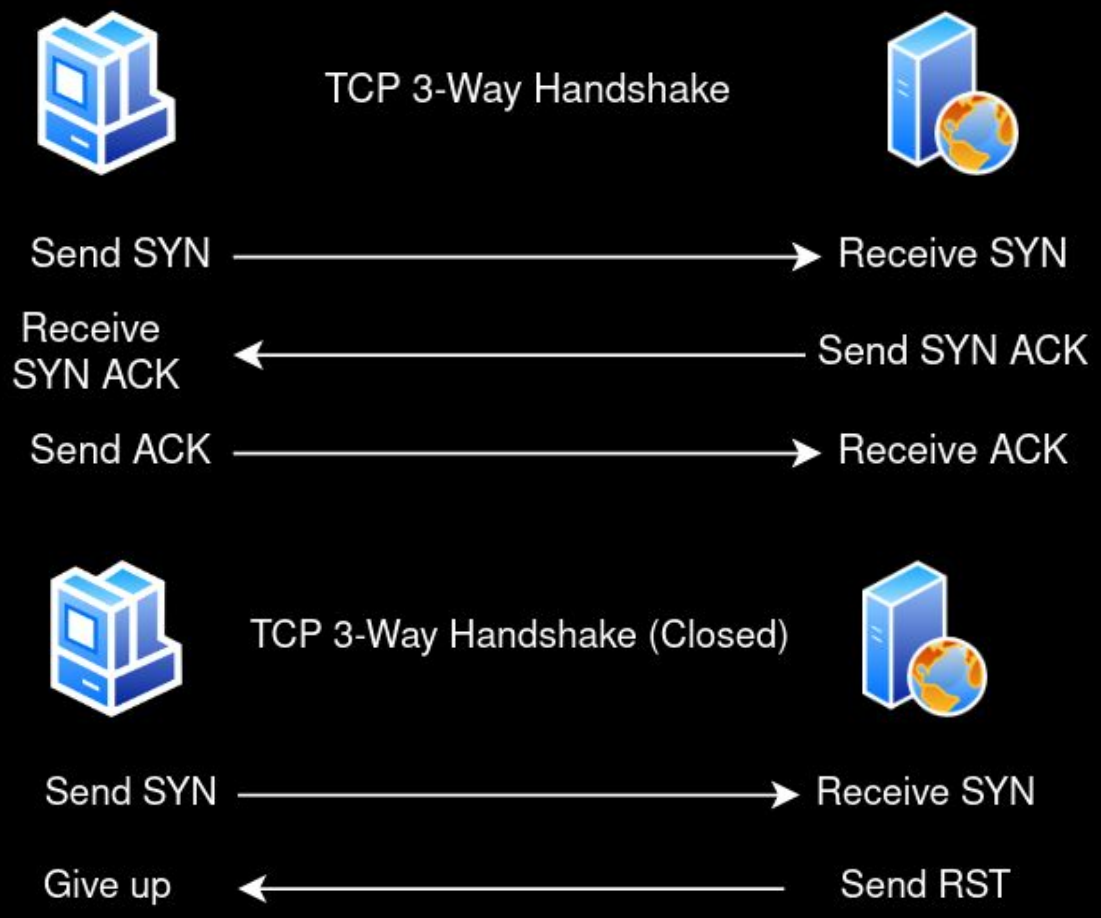
How do we tell **which services** are running on **which ports** for a given IP?

# Transmission Control Protocol



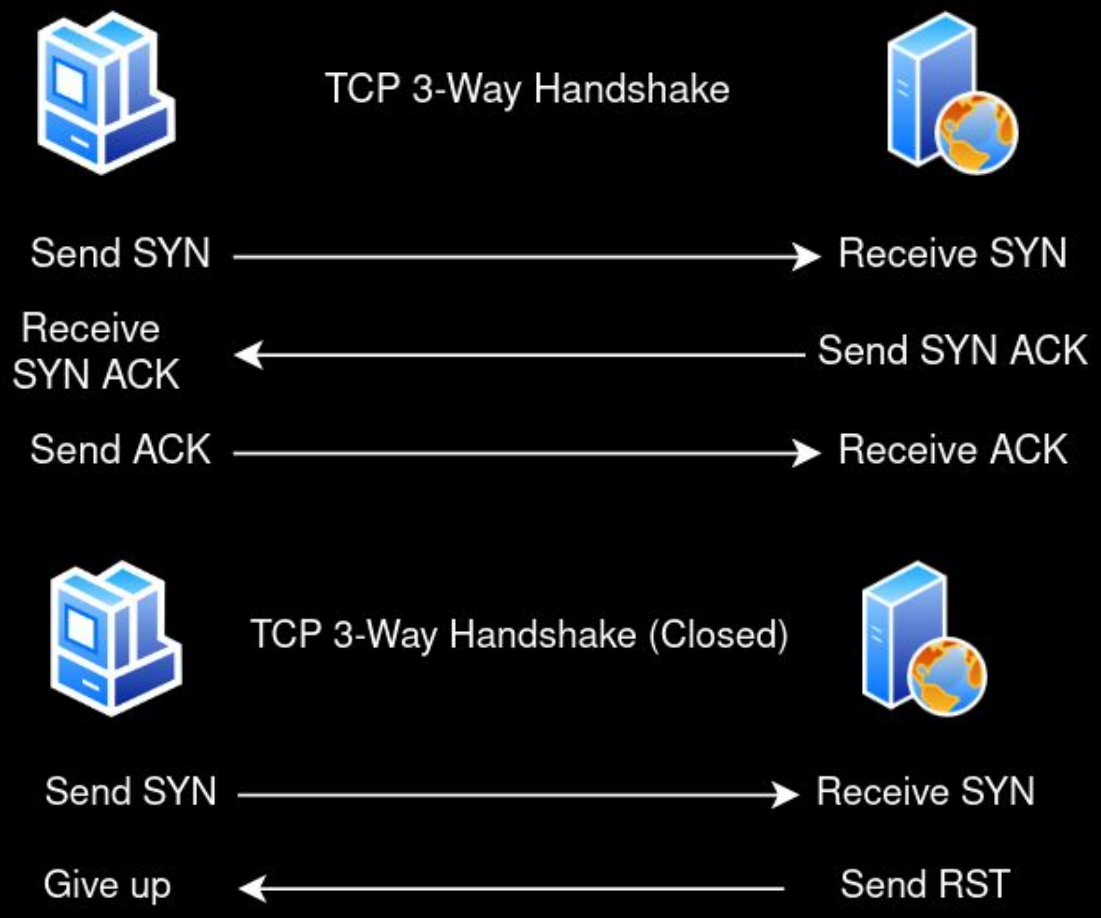

How do we tell **which services** are running on **which ports** for a given IP?

# Transmission Control Protocol




- 65536 TCP ports
- What if we do this handshake on port 0, 1, 2, up to 65535?

# Transmission Control Protocol




- 65536 TCP ports
- What if we do this handshake on port 0, 1, 2, up to 65535?
- **Easy to detect**, but we can randomize the order
- SYN scan: just send the SYN and see if we get anything back (bypasses firewalls!)

# Port Scanning

- This process is **port scanning**, the most important active recon technique
- Port scanning tells us which **ports** are open for a given **IP**
- Once we know this, we can then check each port to see what **protocol** they are talking
- Once we know that, we can check them by protocol to see what **service** they are offering
- Optionally, we can check each **service** by cross-referencing its version with a large database of all known vulnerabilities

# Disclaimer: Passive Recon

- When engaging a real target, there is a lengthy **passive recon** phase before this
- You would scope out owned IPs, domains, employees, and tech stack
- Advanced adversaries would also do their covert infrastructure and malware preparation here
- None of these are relevant for this point in the year

# In Practice

# Port Scanning

- Port range: 0-65535, TCP & UDP
- sudo nmap -Pn -F -sV -vv $IP -oN fast.txt
- sudo nmap -Pn -A -sV -p- -vv $IP -oN full.txt
- -Pn skips the ping check (Windows does not respond)
- -A means that nmap will run scripts and OS fingerprinting (Aggressive)
- -sV will have the scan perform version checking
- -p- will scan every single port from 1-65535
- -vv  will enable very verbose output
- -oN saves the result to a text file so you don't re-scan

# Port Scanning - Edge Cases

- Don't forget UDP services like SNMP!
- `sudo nmap -Pn -F -sU -vv $IP -oN udp.txt`
- `-sU` will have the scan check UDP ports
- `-F` will scan top 1000 ports (UDP scanning is **SLOW**)
- If you're scanning through a SOCKS proxy, you can only scan TCP ports, and should use the **-sT** flag
  - This does a TCP scan rather than SYN scan
- If you're in a network, do a very fast scan using IP range
- General workflow tip: make a directory for each target

# Port Scanning Alternative - Rustscan

- `rustscan` is a modern, insanely fast alternative to **nmap**
- Can scan all 65535 TCP ports in **as fast as 3 seconds**
- **Not stealthy AT ALL**, does not bypass firewalls
- Great for situations where the only thing that matters is speed
- Integrates with nmap for service scanning and script execution
- Generally fewer features

We recommend rustscan for practice like hackthebox!

# Service Scanning: SMB

- Server Message Block runs by default on all Windows computers
- If you know the password, you can view remote file shares
- If the target is running Windows Server or is AD joined, and you have Administrator credentials, **remote code execution is a feature**
- Windows computers prior to Windows 7 SP 6.1 are vulnerable to MS17-010 (**SYSTEM** Remote Code Execution)
- Depending on the target configuration, you can potentially read/write files

# Service Scanning: Other services

- FTP: can be used to upload files or download sensitive files if left unsecured
  - This is especially potent if chained with a web server w/LFI vuln
- SSH: if you have a password or key, you can login and get a shell
- Telnet: like SSH, but without the secure part (yikes)
- SNMP: Simple Network Management Protocol, allows viewing all of the running processes, usernames, and software versions, including command-line arguments (UDP port 161)
- SMTP: Simple Mail Transfer Protocol, runs email server
- MSSQL: Microsoft SQL server, can sometimes **run commands**
- Redis: Database, can **gain RCE as a feature**

# Service Scanning

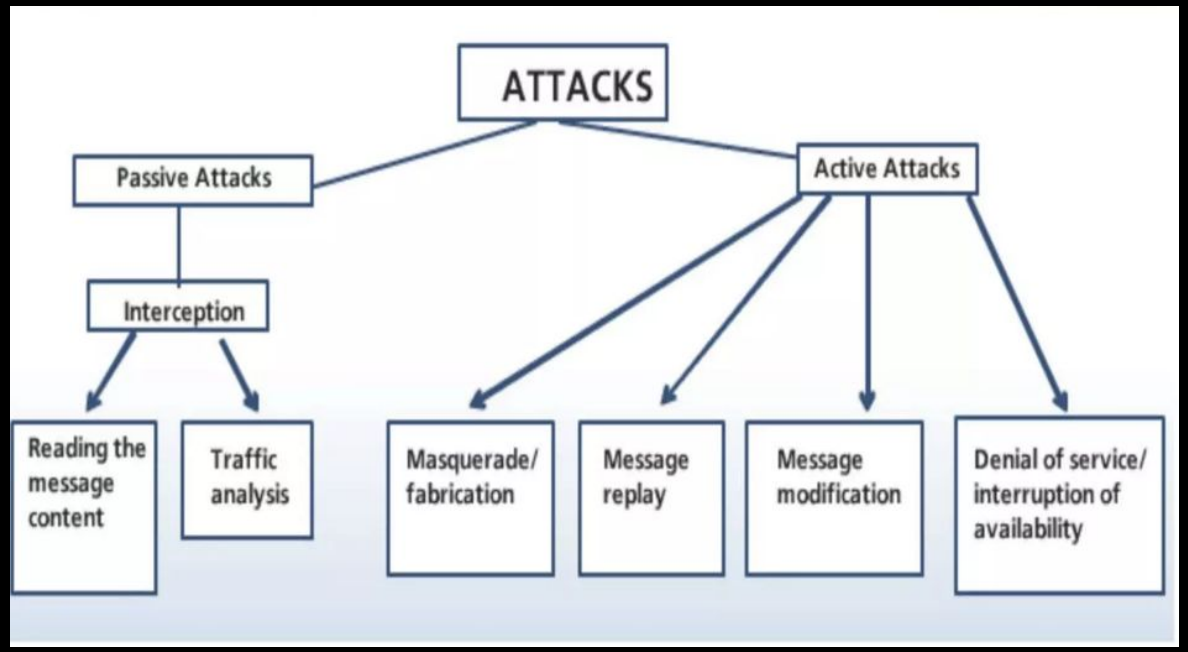
- You won't know every service
- Get in the flow of understanding unfamiliar services quickly and think in terms of primitives (what does the service let me do)
- https://book.hacktricks.xyz/ has some good preliminary steps for interacting with and attacking unfamiliar network services
- Other really important or common services (like web servers & active directory) will be covered individually
- It is very common to see new and unfamiliar services when attacking a network

When you don’t know something, Google it!

# Network Attacks

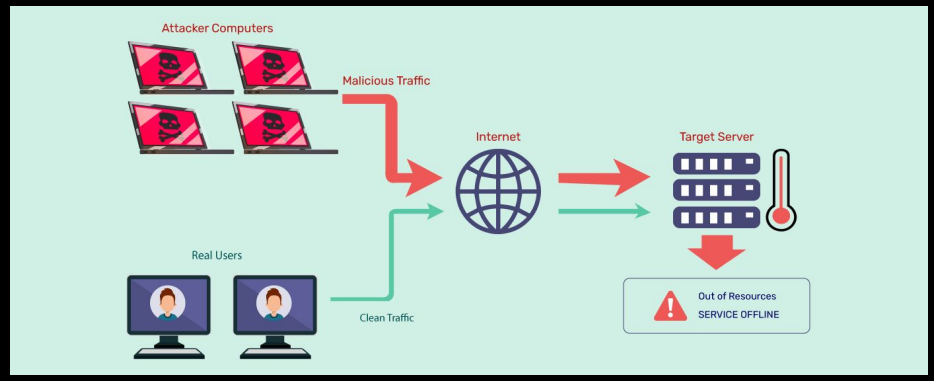
# Types of Attacks

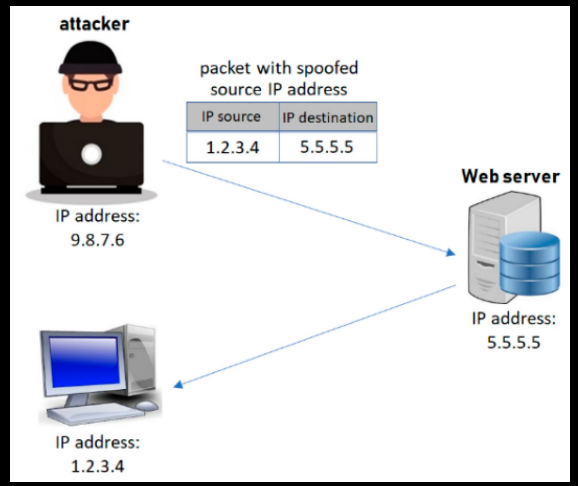
- Passive Attack: Tries to read or make use of information from the system, but doesn't influence operation
- Active Attack: Tries to change the system resources or affect operation

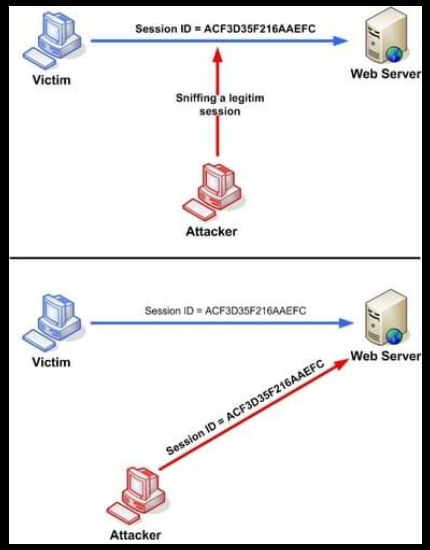



# Some Attacks

- DoS (Denial of Service) Attack (+DDoS)
- IP Spoofing
- Man-in-the-middle Attack
- TCP Session Hijacking



# DoS/DDoS Attacks

- OSI Layer: Transport (L4) / Network (L3)
- What: Aimed at shutting down a network, service, or machine, making it inaccessible to its intended users
- Why: To disrupt business operations, cause financial damage, or cover up other activities by taking critical network services offline
- How: The attacker overwhelms the target with a massive flood of traffic or sends malicious data that triggers a system crash.
- Real-World Incident (2018 Github Attack): Attackers hit GitHub with 1.35 Tbps of incoming traffic. They abused exposed Memcached servers on UDP port 11211, spoofing GitHub's IP to multiply traffic 50,000x. GitHub stayed down for ~8 minutes

# IP Spoofing

- OSI Layer: Network (L3)
- What: Creation of IP Packets with modified source address to impersonate another system
- Why: To bypass IP-based authentication controls, hide the attacker’s true identity, or execute reflection-based DDoS attacks
- How: An attacker uses specialized tools to alter the IP header of outbound network packets, setting the source address field to the IP of a trusted host rather than their own
- Real-World Incident (2016 Dyn Cyberattack): Attackers utilized IP spoofing alongside the Mirai botnet to flood Dyn's DNS servers. By causing major global outages across platforms like Twitter, Netflix, and Reddit.

# Man-in-the-Middle (MitM) Attacks

- OSI Layer: Data Link (L2) / Network (L3)
- What: Intercept and relays communications between parties who believe they are communicating directly
- Why: To eavesdrop on sensitive data or alter the communication payload in transit
- How: The attacker poison the ARP cache (ARP Spoofing) or abuses DNS routing to position their machine in the network traffic path between the victim and the gateway
- Real-World Incident (2015 Lenovo Superfish Incident): Lenovo pre-installed adware called Superfish on laptops. It installed a universal root certificate that intercepted HTTPS traffic. Hackers on the same local Wi-Fi could forge SSL certificates and view encrypted credentials passing through infected laptops.

# TCP Session Hijacking

- OSI Layer: Transport (L4)
- What: An attacker takes control of an active TCP state sequence between two established hosts
- Why: To bypass password-based authentication by seizing an already-authenticated session
- How: The attacker predicts or sniffs the next TCP Sequence Number (SEQ) and Acknowledgment Number (ACK), injects tailored packets into the stream, and desynchronizes the legitimate client
- Real-World Incident (Kevin Mitnick / Tsutomu Shimomura Breach): Kevin Mitnick spoofed a trusted machine's IP and guessed the TCP Sequence numbers of Tsutomu Shimomura's server, allowing him to hijack an unencrypted shell connection without credentials.

# Application-Layer Threats

1. SQL Injection
   - Injecting raw SQL commands into unchecked input fields to manipulate backend queries
   - Hackers breached Equifax, exposed data of nearly 147M people
2. Cross-Site Scripting
   - Injecting malicious JavaScript payloads executed within a victim's browser session
   - Hackers breached Canvas in 2026, began demanding ransomware and caused downtime
3. Password Brute Force
   - Rapidly testing combinations of usernames and passwords against login endpoints
   - Over 85,000 Fortinet firewall and SSL-VPN gateways globally were compromised by threat actors.

# And Their Network Defenses

1. SQL Injection: WAFs analyze inbound Layer 7 HTTP payloads to inspect parameters and drop packets matching SQL signature patterns before they hit backend servers
2. XSS: WAF Filtering & TLS Interception inspect traffic for script injection patterns (<script>) while enforcing strict Content Security Policies delivered over encrypted network channels
3. Password Brute Force: Network Rate-Limiting & Gateways (e.g., NGINX, API Gateways) enforce connection thresholds per source IP, temporarily blocking IP addresses that generate excessive failed requests within a short timeframe

# Next Meetings

**2025-09-10 • This Thursday**

- Network Security Lab
- We'll put this knowledge into practice by port scanning and getting into systems on HTB

**2025-09-15 • Next Tuesday**

- Getting Your First Reverse Shell
- Once you’re in the system, what next?

ctf.sigpwny.com

`sigpwny{SYN;SYN_ACK;ACK;}`

**Meeting content can be found at sigpwny.com/meetings.**

SIGPwny