SEGSI

CIBERSEGURANÇA E ADMINISTRAÇÃO DE SISTEMAS

DCR

Why security?

- Hot topic!
- Compromises are increasing. Breaches of data affects not only enterprises but real people i.e.:
  - Equifax
  - Portal Base
  - TAP
  - Revolut - https://cybermagazine.com/cyber-security/revolut-hacked-as-cyber-criminals-steal-us-20m

What are we going to takle here

- Hands-On work
- Hardening
- Architecture/Network Security Controls
- Data Encryption
- Security Monitoring

Experience required

- Wireshark
- System Administration (Windows & Linux)
- DevOps Experience
- Networking Concepts
- Kubernetes/Container Orchestration

Where to get that experience

Wireshark

https://www.amazon.es/-/pt/dp/1893939758

System Administration (Windows & Linux)

https://www.amazon.es/-/pt/dp/0134277554

DevOps Experience

Networking Concepts

CCNA https://www.amazon.es/-/pt/dp/1587205807

Kubernetes/Container Orchestration

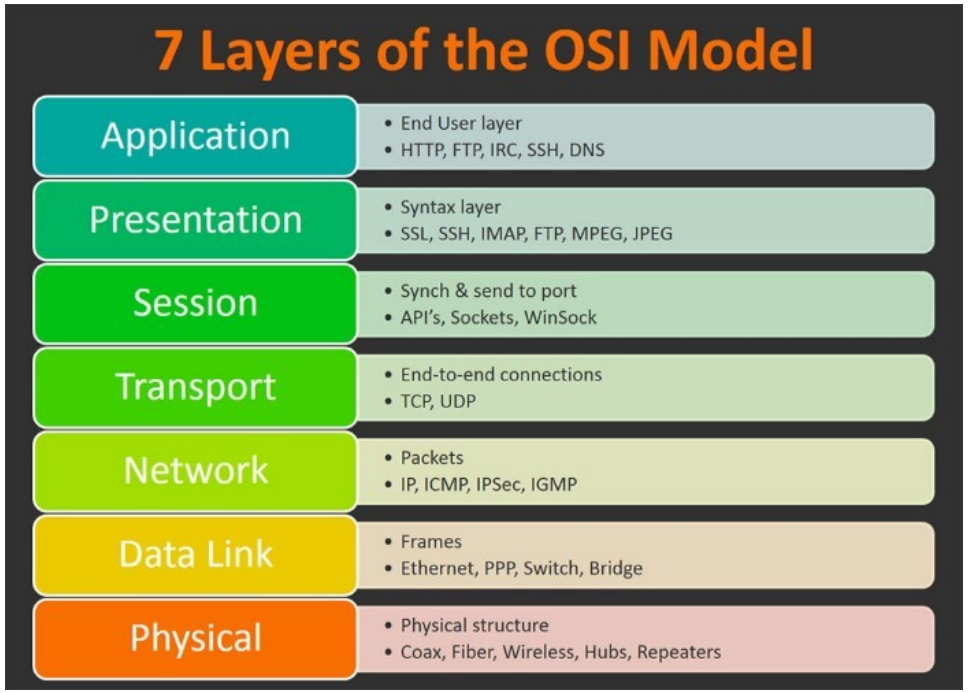
https://www.amazon.es/-/pt/dp/1839211253/

https://docs.cilium.io/en/stable/

Network Stack

- The OSI model is a conceptual model that tries to limit each layer to a specific function
- Each layer is interchangeable and generally do not affect others
- Each layer can contain a subset of fields

Open Systems Interconnection (OSI) Model

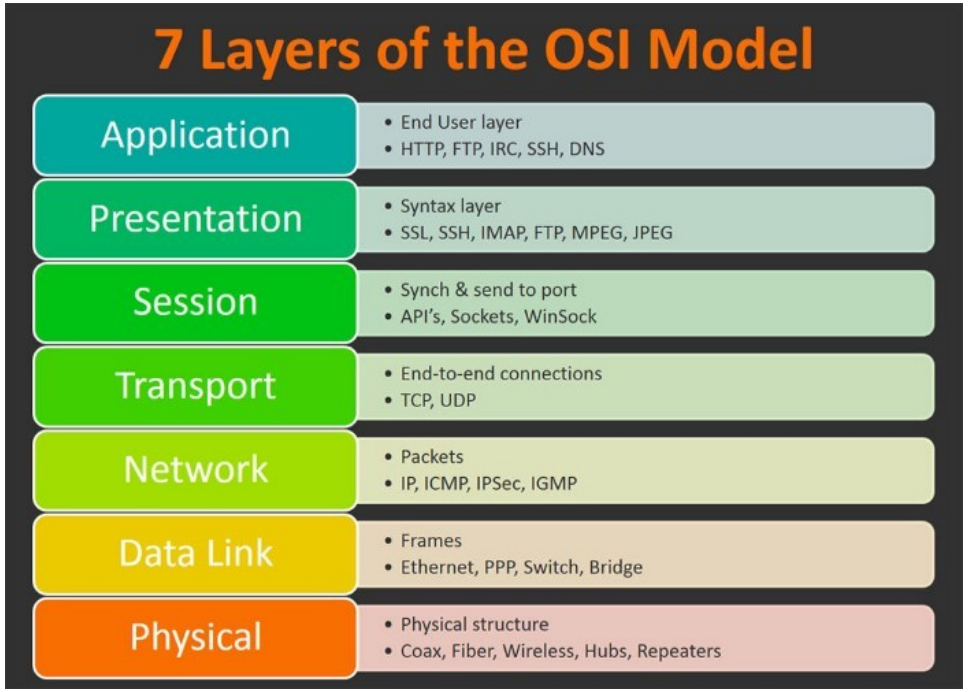


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Physical Layer

- The physical layer is the medium where the data is transmitted
- Several types exist such as:
  - Ethernet Cable
  - Air
  - Fiber
- Attacks:
  - Traffic Interception on Cable connections

OSI Model

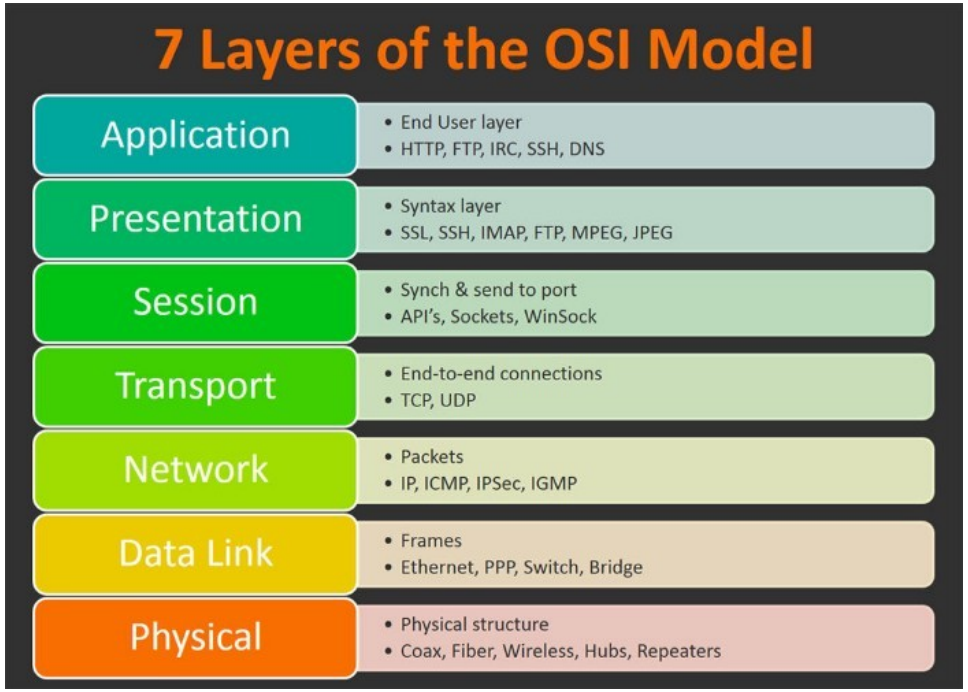


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Data Link

- The data link is often related to the local network
- **Frames** are the unitary pieces of information flowing on this layer
- Protocols for the local network reside on this layer such as **ARP**
- Attack
  - ARP Spoof

OSI Model

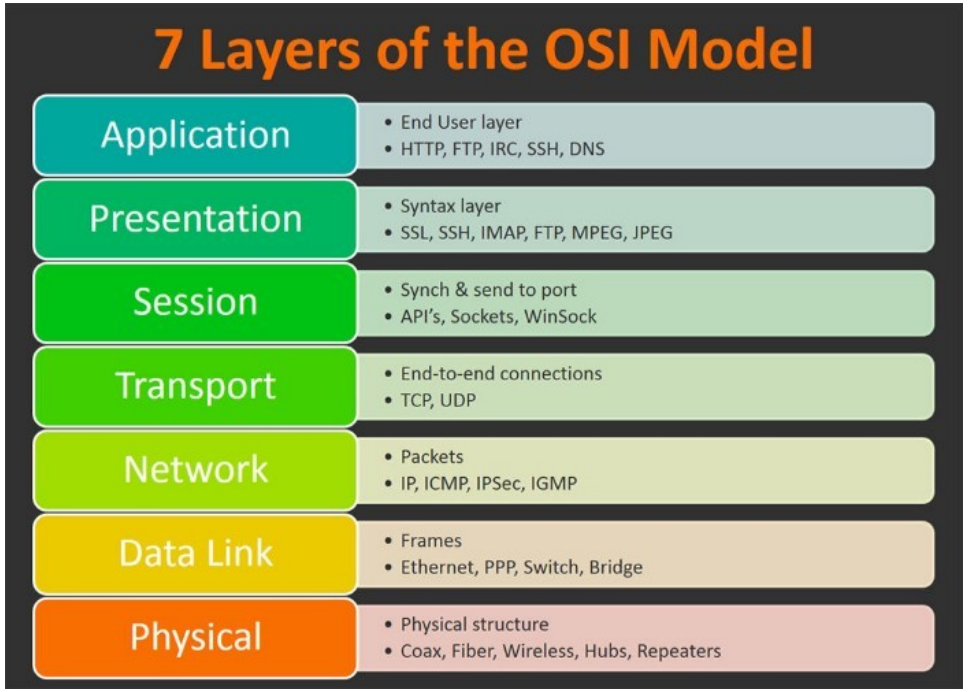


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Network

- The network layer considers **IP packets**
- It allows for internetwork communication protocols such as IP and routing protocols
- Attack
  - BGP Hijack

OSI Model

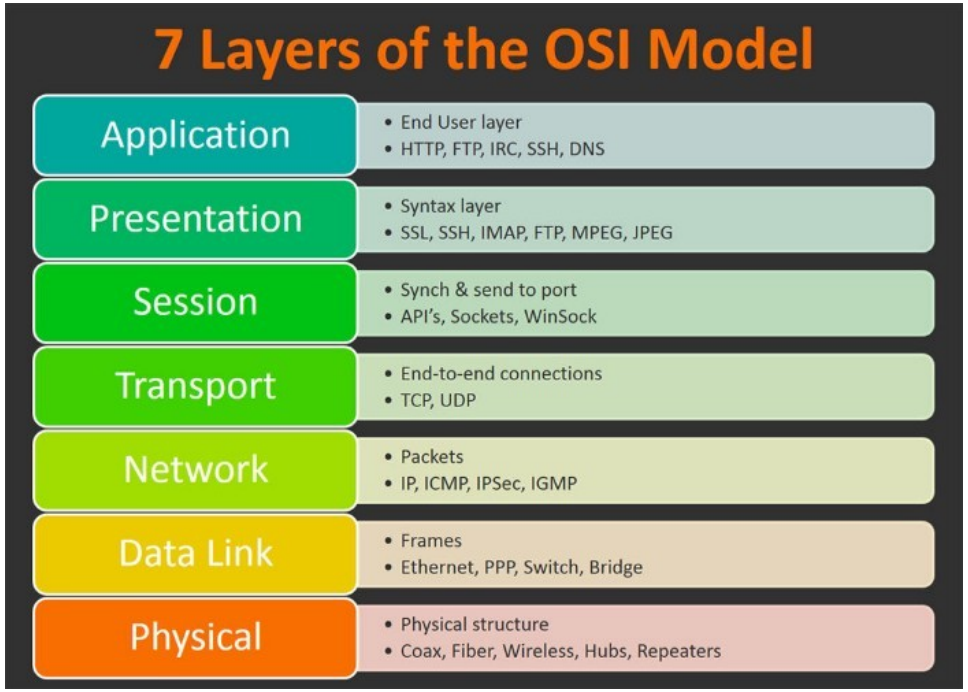


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Transport

- Describes the mechanism to transport data
  - **TCP** - order and reliability matters, heavier
  - **UDP** - size and performance matters, lighter
- Attack
  - DrDoS
  - TCP Session Stealer

OSI Model

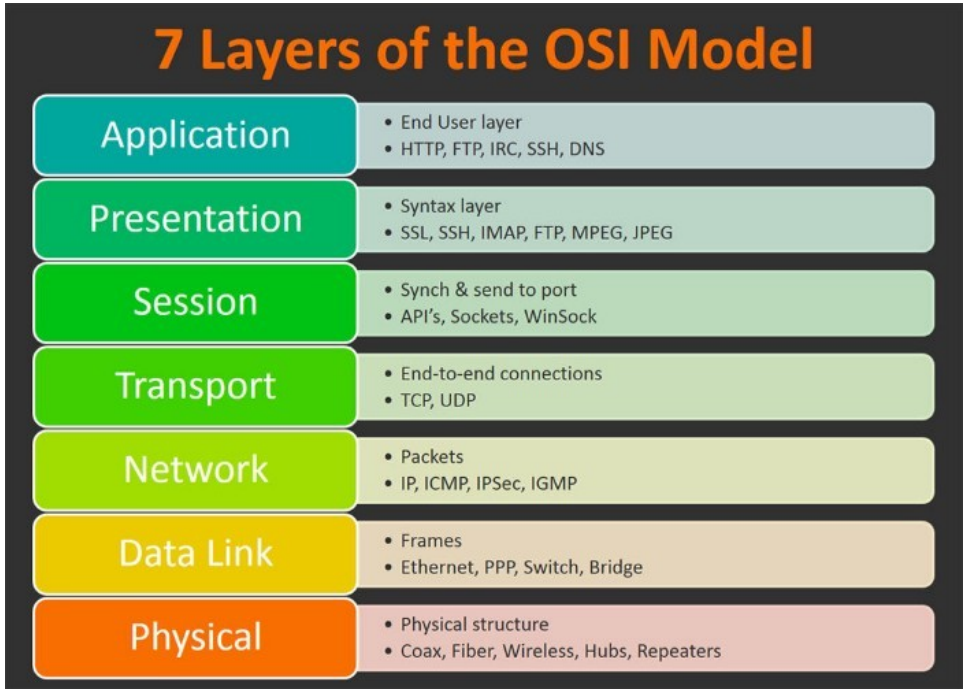


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Session

- Keeps track of sessions in the machine
  - Some protocols use this layer primarily to ensure that operations are unitary, and order is maintained
- Attack
  - Session Stealers
  - IPC hijack

OSI Model

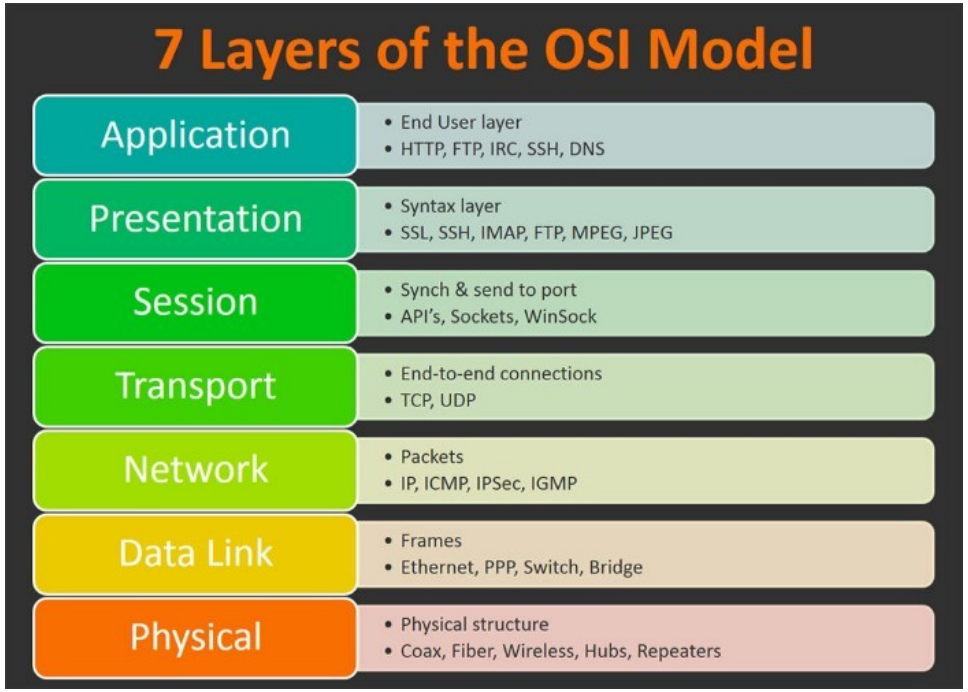


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Presentation

- Handles the data presentation in case of needed conversion
  - Encryption/Decryption
  - Compressing/Uncompressing
- Attack
  - Buffer Overflow (CoreFTP CVE-2020-19595)

OSI Model

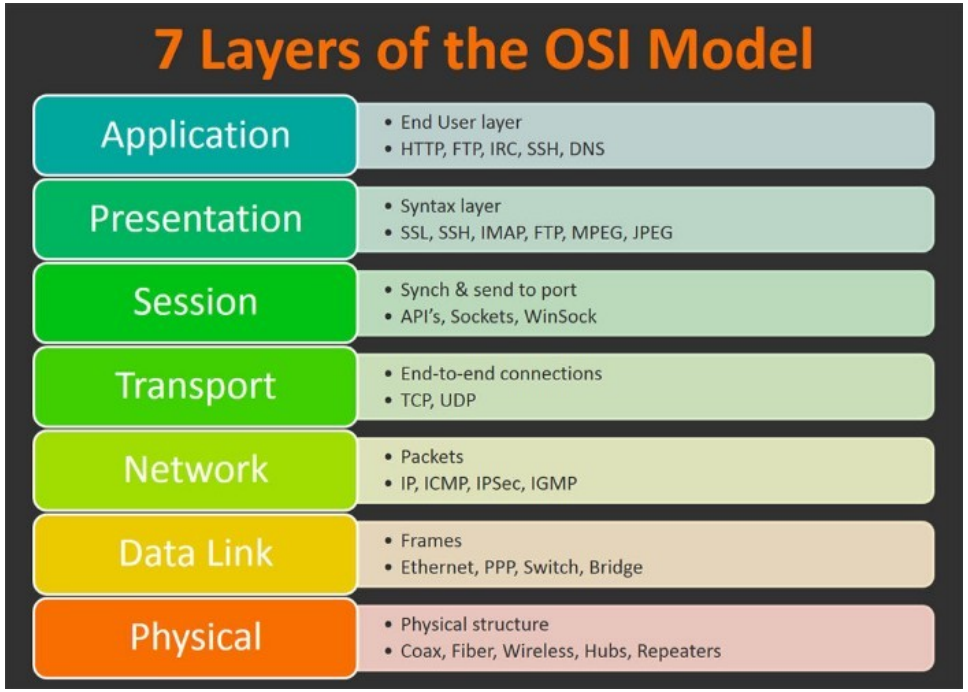


https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Network Stack - Presentation

- End application that will receive and process information to the user, for instance, **HTTP** as you browse the Internet
- Attack
  - Application Attacks (SQL injection, XSS, others..)

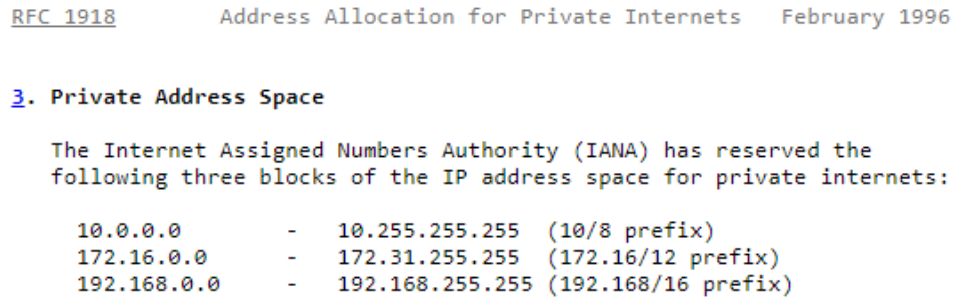
OSI Model



https://int0x33.medium.com/day-51-understanding-the-osi-model-f22d5f3df756

Local Area Network

- A network interface can hold several IPs
  - These are its identifiers on the network
- An **IP address** is a 4-byte host address in dotted-decimal notation
  - **123.123.123.123**
  - **This converts to 0x7B7B7B7B in the program memory**
- Usually, a Local Area Network uses the RFC1918 as address scheme
  - https://datatracker.ietf.org/doc/html/rfc1918



RFC 1918 Address Allocation for Private Internets February 1996

3. Private Address Space

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

10.0.0.0	-	10.255.255.255	(10/8 prefix)
172.16.0.0	-	172.31.255.255	(172.16/12 prefix)
192.168.0.0	-	192.168.255.255	(192.168/16 prefix)

Local Area Network

- An IP address is not a network, we need to define a rule to identify our network
- Let 192.168.100.1/24 be IP address

- Q-What is the /24?
- A-It's the Class Inter Domain Routing prefix (CIDR) an abbreviation of the Network Mask

- Q-What is the network mask?
- A-A mask consisting of all 1's aligned left that, by doing a logical AND operation with the host address, will result in the network address.

Local Area Network

Then:

192.168.100.1 (IP address)

AND 255.255.255.0 (Netmask /24)

192.168.100.0

- 255.255.255.0 ==
- 11111111111111111111111100000000 (binary)

- An invalid mask would be something that has a 1 after a 0 such as
- 11111111111111111111111100001000

Local Area Network

But why?

Because LANs have intrinsically two special addresses:

- The network address
- The broadcast address

These addresses are reserved and not valid for a network interface

- The network address
  - This first IP address in a subnet
  - If 192.168.100.0/24 is our network, then it is the network address
- The broadcast address:
  - The last network address in a subnet
  - If 192.168.100.0/24 is our network, then 192.168.100.255 is our broadcast address

  Note: 255==0xFF (1 byte), that's the maximum

Local Area Network

Again, why?

To discover devices in our network of course. We can segment networks to better use the IP address space and manage the network.

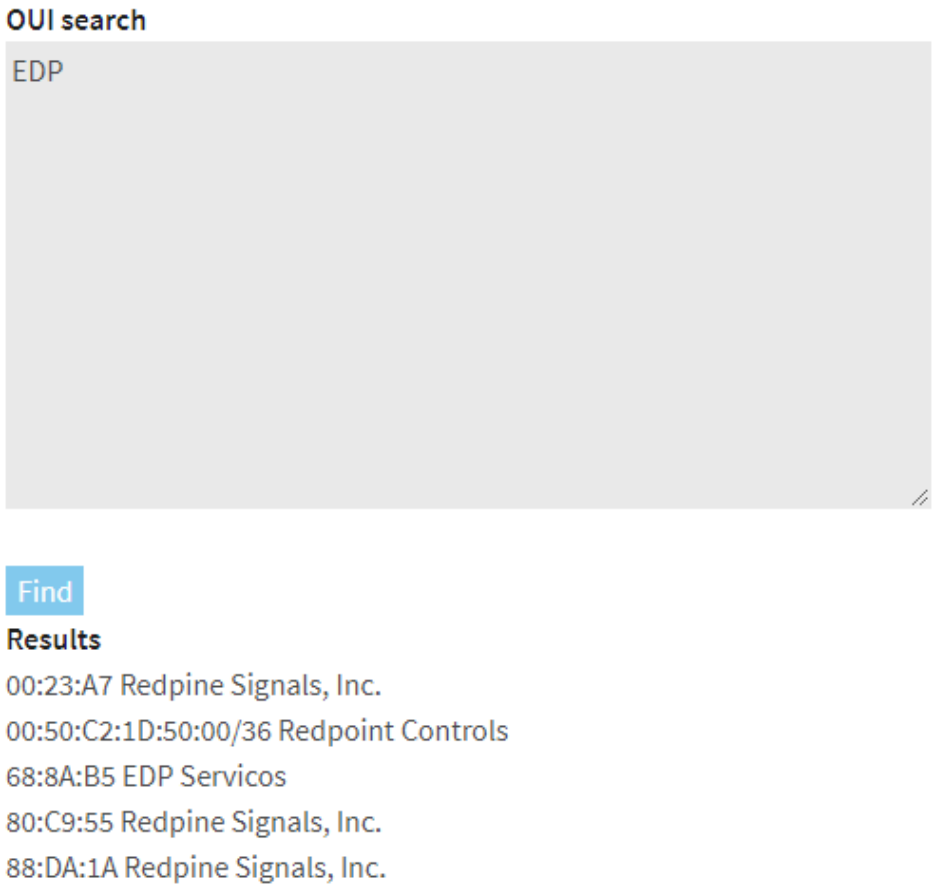
In a network we don’t know our neighbors. In local networks we need to translate an IP address to a MAC address.

Queue **ARP…**

Local Area Network - ARP

ARP - Address Resolution Protocol is the protocol that given an IP address will translate it to a MAC address

MAC (address) - **Media Access Control** address is a 48 bits address “unique” per device (we can change this easily) where half of it determines the device manufacturer and the remaining is the interface “ID”



https://www.wireshark.org/tools/oui-lookup.html
- Check EDP devices on this website.

Local Area Network - ARP

- Computer A wants to send data to Computer B in the same local network, how does this proceeds?
  - Computer A sends an ARP resolution request to the broadcast address
  - All the network devices receive the packet, but computer B sees that it is meant to it and replies
  - Computer B sends the reply directly stating: If you want to reach me use the following MAC address
  - Computer A receives the reply and stores the information on its **ARP table**, it then proceeds making the necessary connection using the MAC address

Wide Area Network (WAN)

- What if we want to reach devices outside our own local network?
- A device in our network must connect to a Wider network (remember the WAN port on your router?)
- We then must send traffic to that device
- Essentially:
  - Is the traffic to our local network -> use direct link
  - Is the traffic to another network -> use a **gateway** to **route** packets (this gateway will be our **default route**)

Wide Area Network (WAN)

- A default route/default gateway should be configured
- **ARP** will be used to send packets to the Default Gateway, but the IP address will be the real destination
  - **MAC** on the packet is to the Default Gateway, but the IP address will be a remote network

Ports

- Communications need to maintain a state, so the computer keeps track of what traffic goes to each application
- Operating Systems use the concept of **Ports** to maintain this state
- Ports range from 0-65535 (65536 available ports, or 0xFFFF)
- Some **lower port** numbers **are registered**, meaning that only Administrators can change the service running and usually follows a known service/port, usually from the 0-1024 range
  - HTTP: tcp/80
  - SMTP: tcp/25
  - https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#Well-known_ports

Ports

- This all translates to the following: to interact with a service we must know:
  - The destination IP address
  - The Port where the service is running

Routes

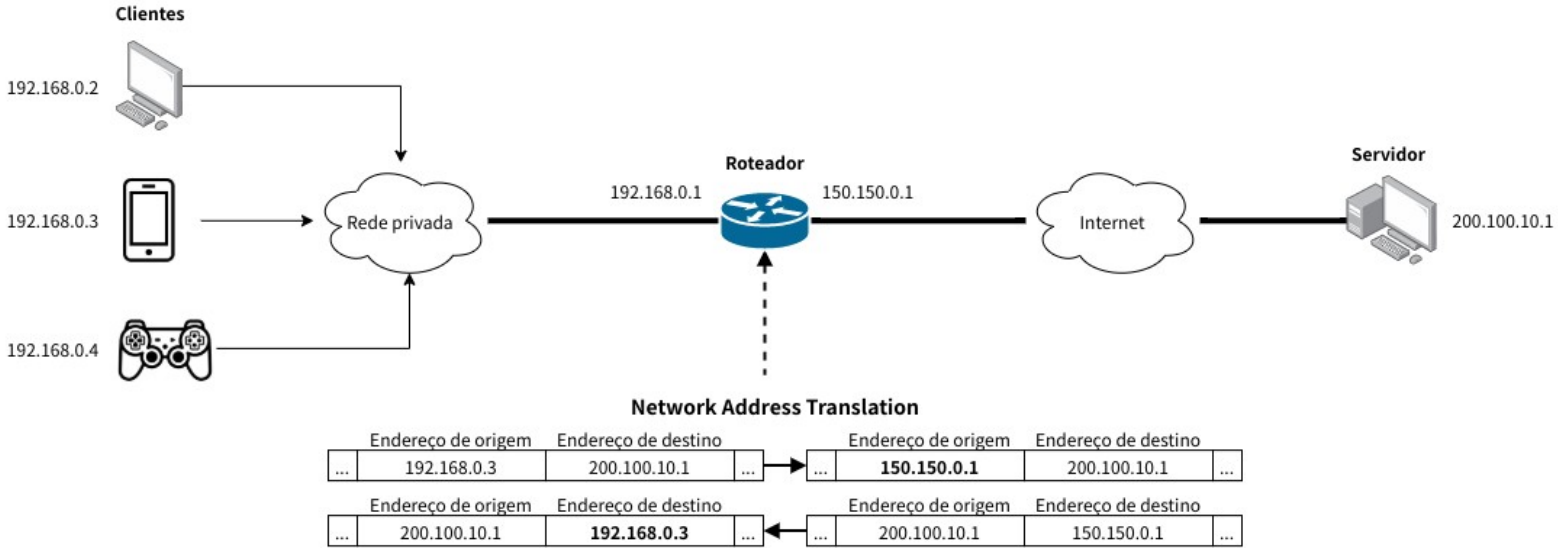
- Like the default route, other routes can be added to different paths
- A router contains several routes to redirect traffic to the specific networks
  - Gigapix routes traffic to other countries https://gigapix.pt/pt/home/
- A route is essentially built by the following:
  - A destination network
  - Next hop to reach that destination
  - Weight (route priority in case it goes down, a fallback link may be used)
- BGP Looking Glass
  - https://lg.he.net/

NAT

- Count the number of network devices (mobile, computers, telephones, …) now do the math on how many IPv4 Address are available to use
- The numbers don’t match up. There are a lot more computers than available addresses
- This is because several computers share the same **RFC1918** address space, but then their communications go through an outbound router with one IPv4 address. The operation is call Network Address Translation.

NAT

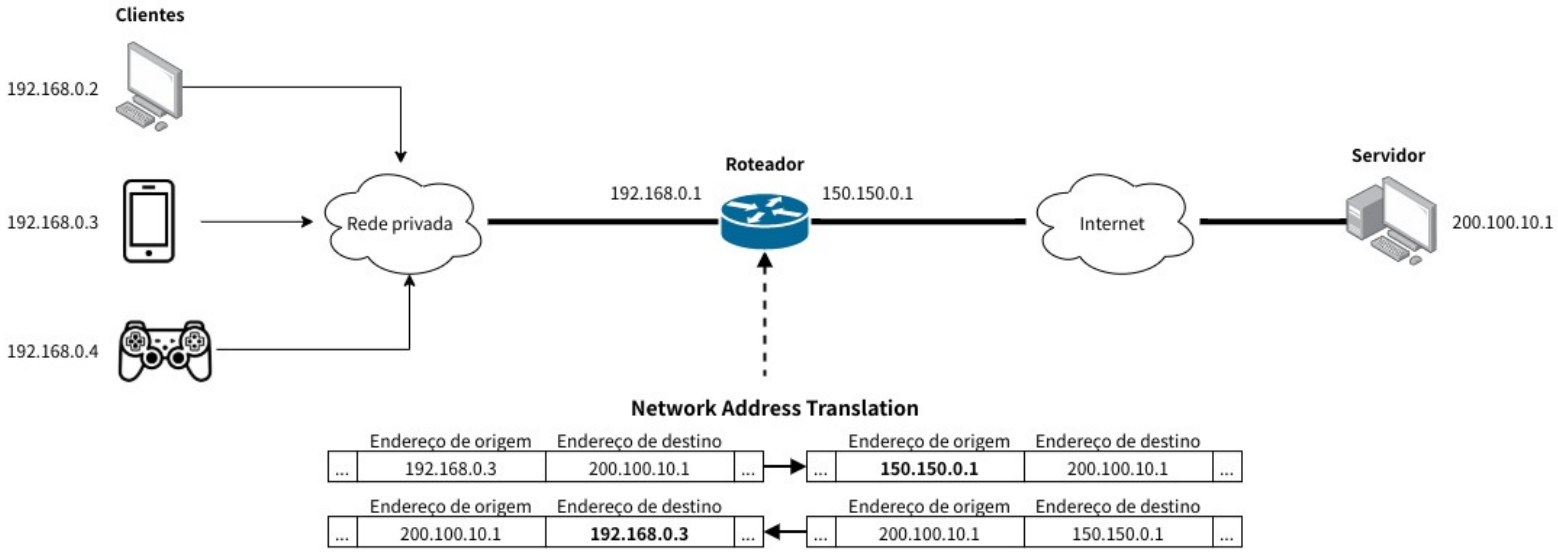
- **Network Address translation**, or NAT for short is the process that will convert a flow from an internal PC to the Internet, allowing multiple computers to reuse the same Public IP address
- **Home routers** do this by default



https://pt.wikipedia.org/wiki/Network_address_translation#/media/Ficheiro:Network_Address_Translation_(NAT).png

NAT

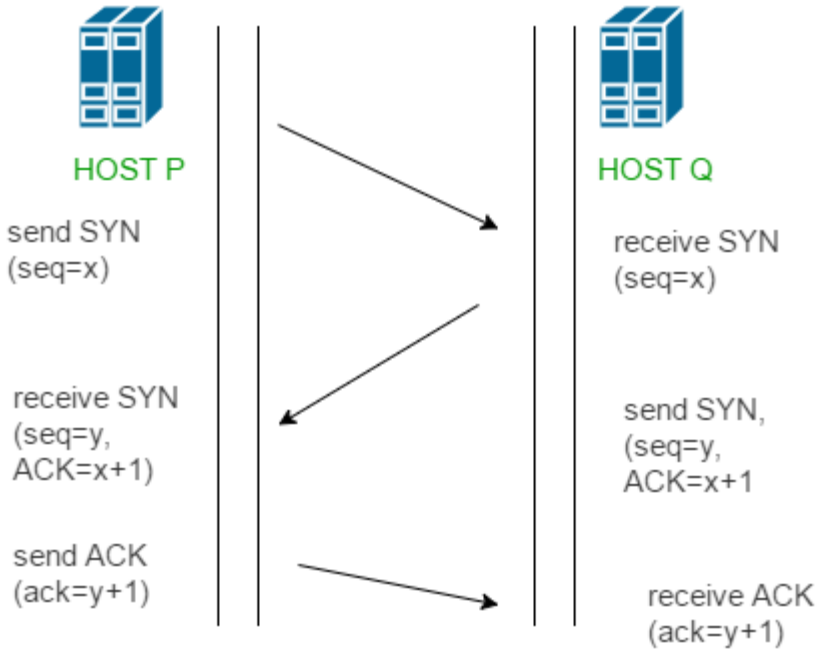
- **NAT** will replace the source address when sending a packet trough the Internet and convert back when receiving it.



https://pt.wikipedia.org/wiki/Network_address_translation#/media/Ficheiro:Network_Address_Translation_(NAT).png

TCP

- https://datatracker.ietf.org/doc/html/rfc7414
- Agreement of client and server to establish a connection
- Guarantees order and delivery (otherwise an error)



https://www.geeksforgeeks.org/tcp-3-way-handshake-process/

Linux Network Tools

- **ip link** – Get Status, add or remove of network interfaces
- **ip addr** – Add/remove IP addresses of network interfaces
- **ip route** – Add/remove routes
- **arp** – Manage the computer ARP table
- **ss** – Sockets open in the machine
- **whois** – Check Ownership of an IP address

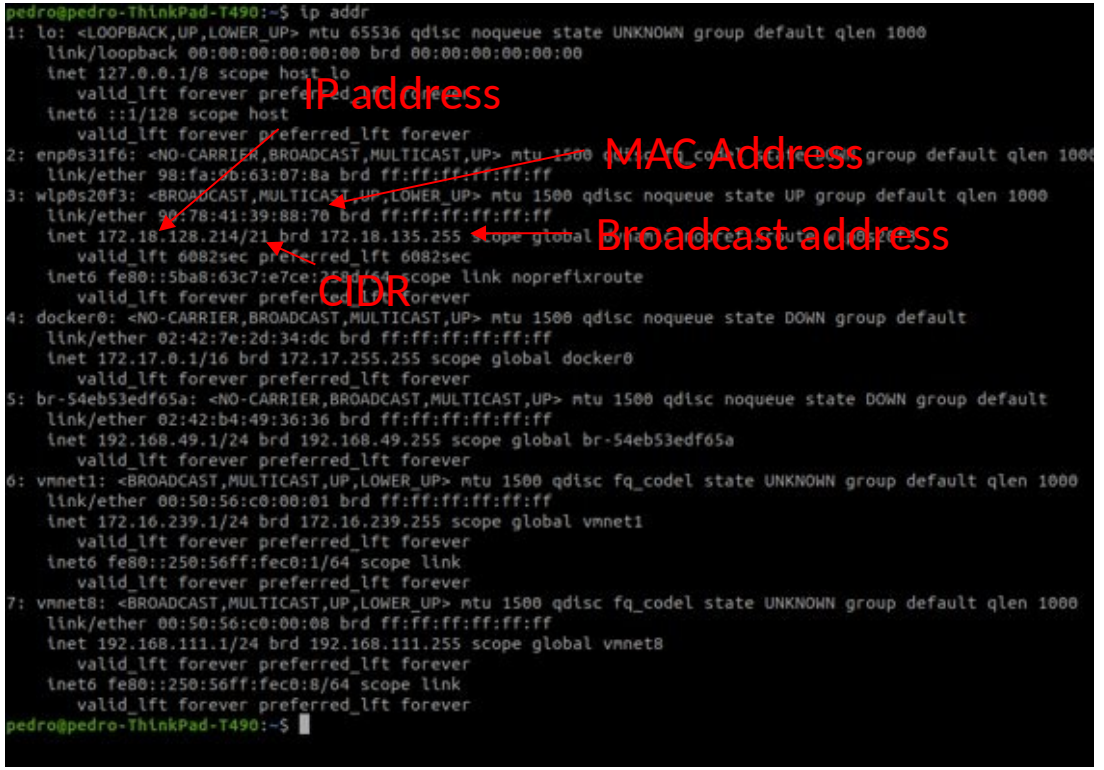
Linux Network Tools

- **ip link**

```
pedro@pedro-ThinkPad-T490:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s31f6: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN mode DEFAULT group default qlen 1000
    link/ether 98:fa:9b:63:07:8a brd ff:ff:ff:ff:ff:ff
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DORMANT group default qlen 1000
    link/ether 90:78:41:39:88:70 brd ff:ff:ff:ff:ff:ff
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default
    link/ether 02:42:7e:2d:34:dc brd ff:ff:ff:ff:ff:ff
5: br-54eb53edf65a: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default
    link/ether 02:42:b4:49:36:36 brd ff:ff:ff:ff:ff:ff
6: vmnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether 00:50:56:c0:00:01 brd ff:ff:ff:ff:ff:ff
7: vmnet8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether 00:50:56:c0:00:08 brd ff:ff:ff:ff:ff:ff
pedro@pedro-ThinkPad-T490:~$
```

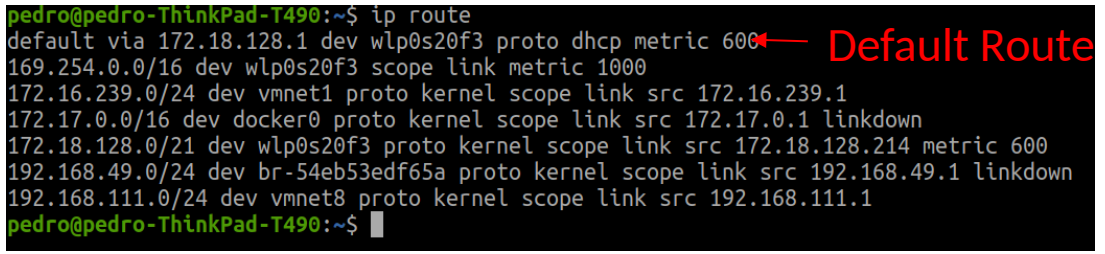
Linux Network Tools

- **ip addr**
  - **ip address add 192.168.1.1/24 dev enp3s0**
  - **ip address del 192.169.1.1 dev enp3s0**



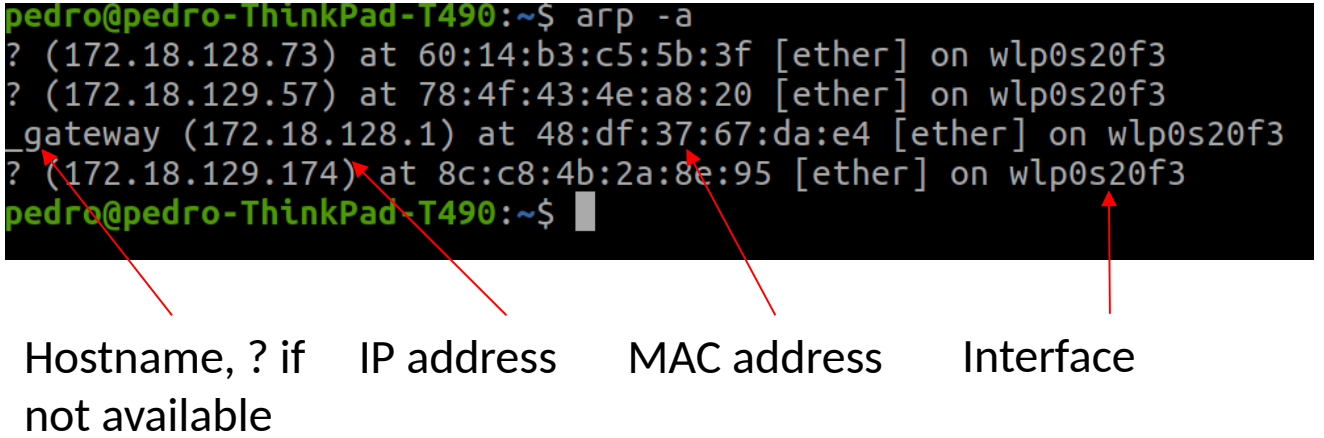
Linux Network Tools

- **ip route**
  - **ip route add 192.168.1.0/24 via 192.168.20.1**
  - **ip route del 192.168.1.0/24 via 192.168.20.1**



Linux Network Tools

- **arp -a** #View the system ARP table
- **arp -d <address>** #Remove entry from ARP table



Linux Network Tools

- **ss -t4ln**
  - -t -> TCP
  - 4 -> IPv4
  - l -> LISTENING
  - n -> do not resolve port numbers

```
pedro@pedro-ThinkPad-T490:~$ ss -t4ln
State         Recv-Q        Send-Q                Local Address:Port                Peer Address:Port       Process
LISTEN        0             4096                  127.0.0.53%lo:53                       0.0.0.0:*
LISTEN        0             4096                      127.0.0.1:17622                    0.0.0.0:*
LISTEN        0             4096                      127.0.0.1:39447                    0.0.0.0:*
LISTEN        0             128                       127.0.0.1:631                      0.0.0.0:*
pedro@pedro-ThinkPad-T490:~$
```

Linux Network Tools

- **whois <address> #View the ownership information of an IP address**
  - **whois 193.136.56.49**

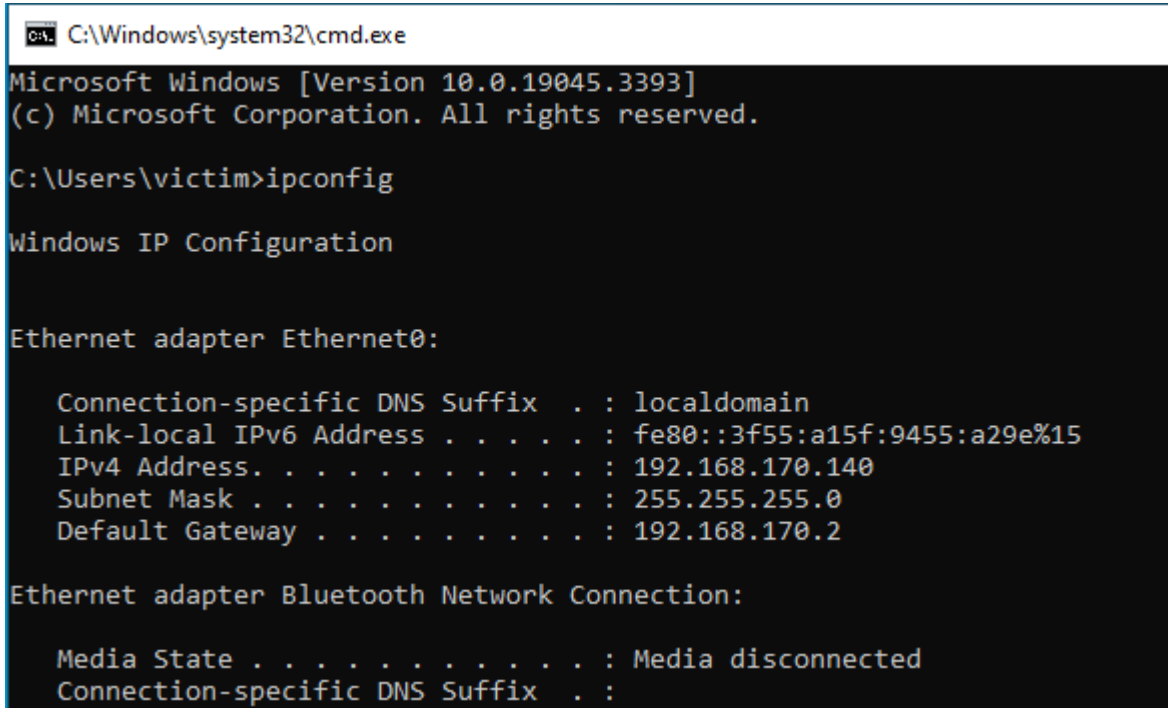
```
pedro@pedro-ThinkPad-T490:~$ whois 193.136.56.49
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '193.136.56.0 - 193.136.63.255'

% Abuse contact for '193.136.56.0 - 193.136.63.255' is 'report@cert.rcts.pt'

inetnum:        193.136.56.0 - 193.136.63.255
netname:        IPP-IDT-NET
descr:          Instituto Politecnico do Porto
descr:          Instituto Desenvolvimento Tecnologico
descr:          Porto
country:        PT
geoloc:         41.179616 -8.596491
admin-c:        RCIP12-RIPE
tech-c:         RCIP12-RIPE
status:         ASSIGNED PA
org:            ORG-IPDP1-RIPE
remarks:        SERVIP-IPP
mnt-by:         AS1930-MNT
mnt-lower:      AS1930-MNT
created:        1970-01-01T00:00:00Z
last-modified:  2017-03-28T09:44:47Z
source:         RIPE
```

Windows Network Tools

- **ipconfig** – Get Status, add or remove of network interfaces
- **netsh** – Network Configuration Shell
- **route** – Add/remove routes
- **arp** – Manage the computer ARP table
- **netstat** – Sockets open in the machine

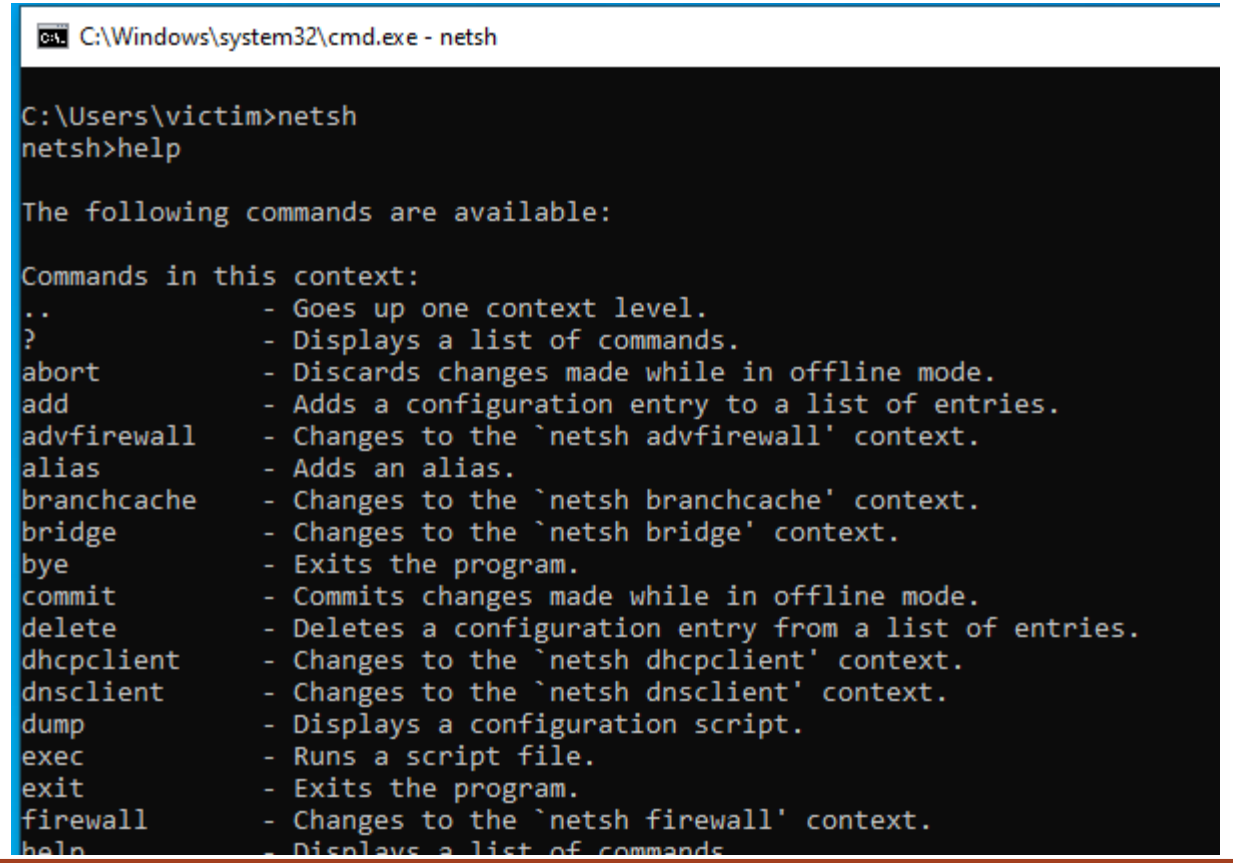
Windows Network Tools

- **ipconfig** – Get Status, add or remove of network interfaces



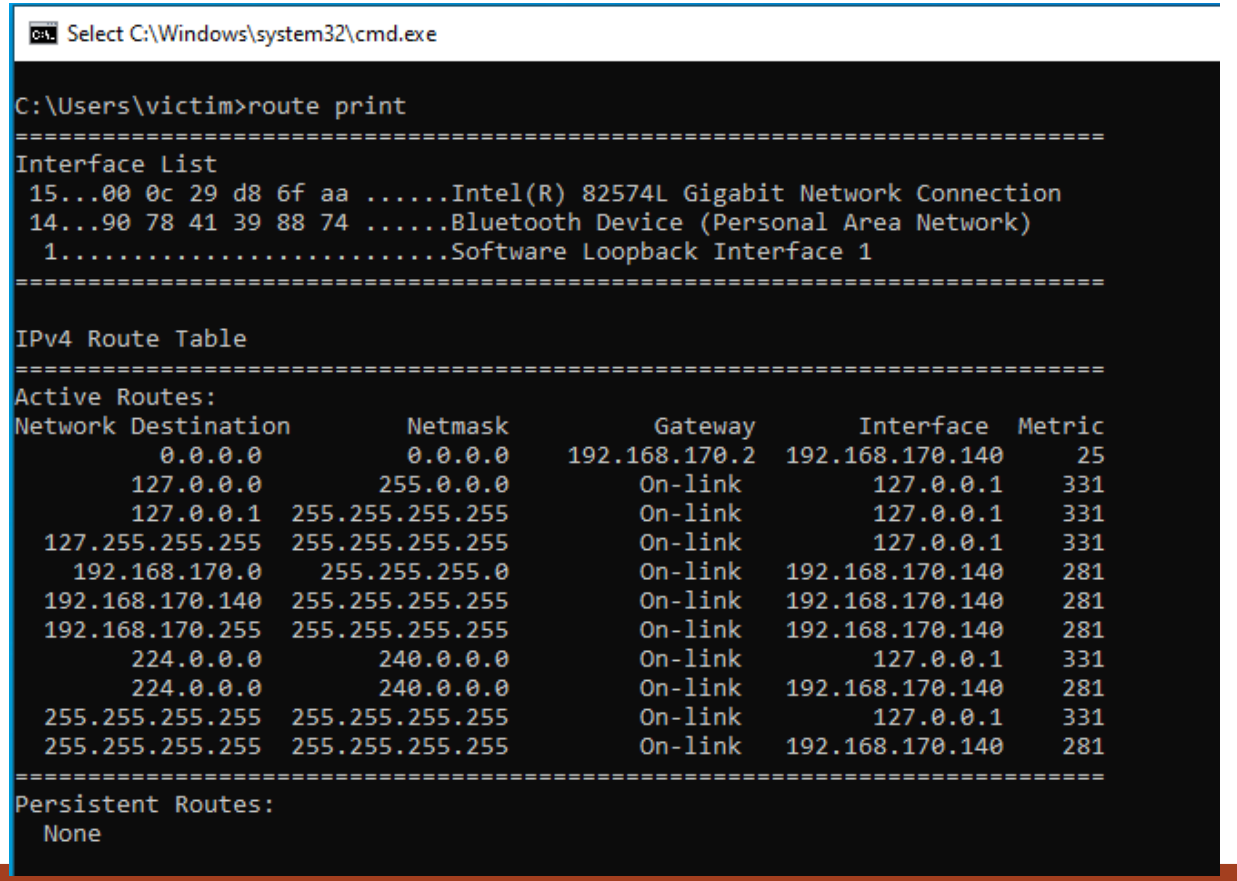
Windows Network Tools

netsh – Network Configuration Shell



Windows Network Tools

- **route** – Add/remove routes



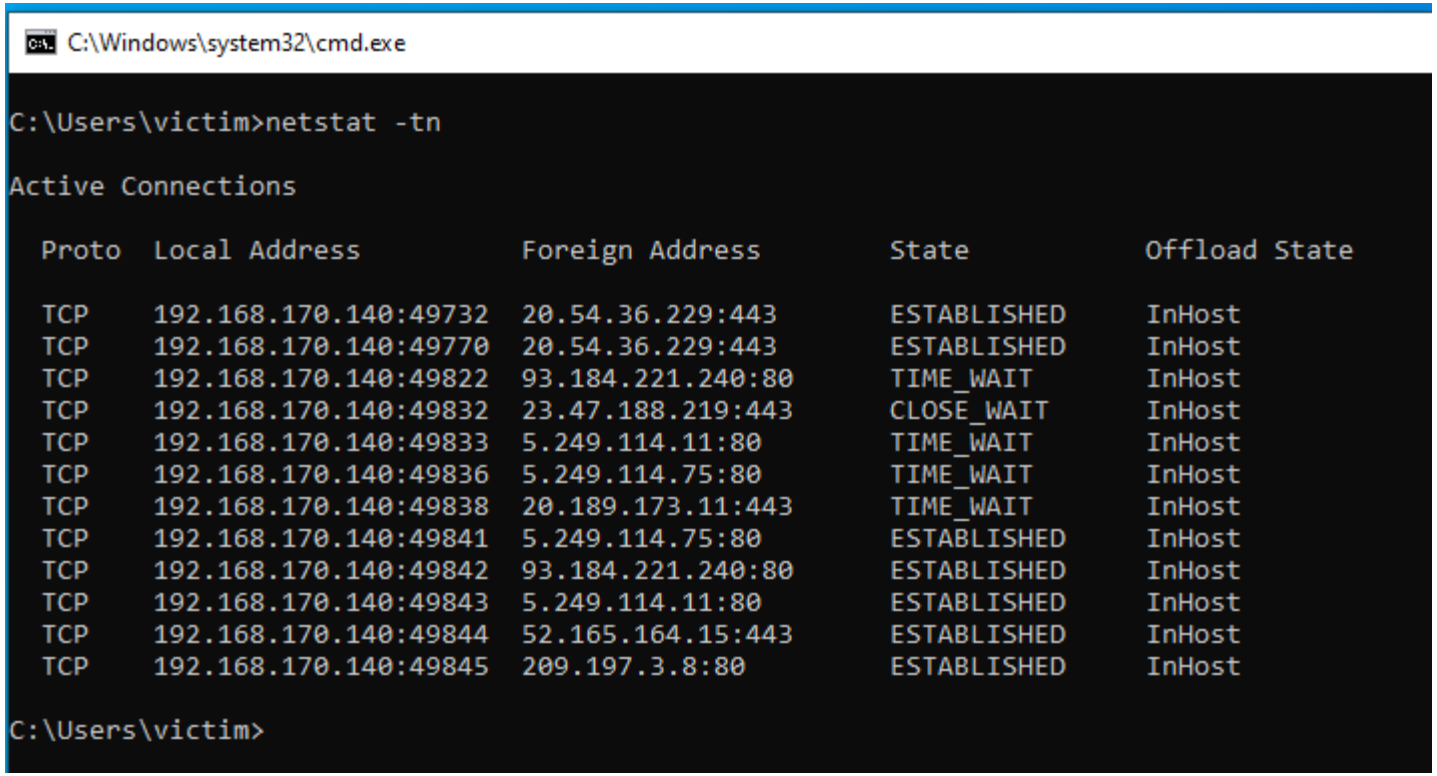
Windows Network Tools

- **arp** – Manage the computer ARP table

```
C:\Users\victim>arp -a

Interface: 192.168.170.140 --- 0xf
  Internet Address      Physical Address      Type
  192.168.170.2         00-50-56-ef-0e-c1     dynamic
  192.168.170.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

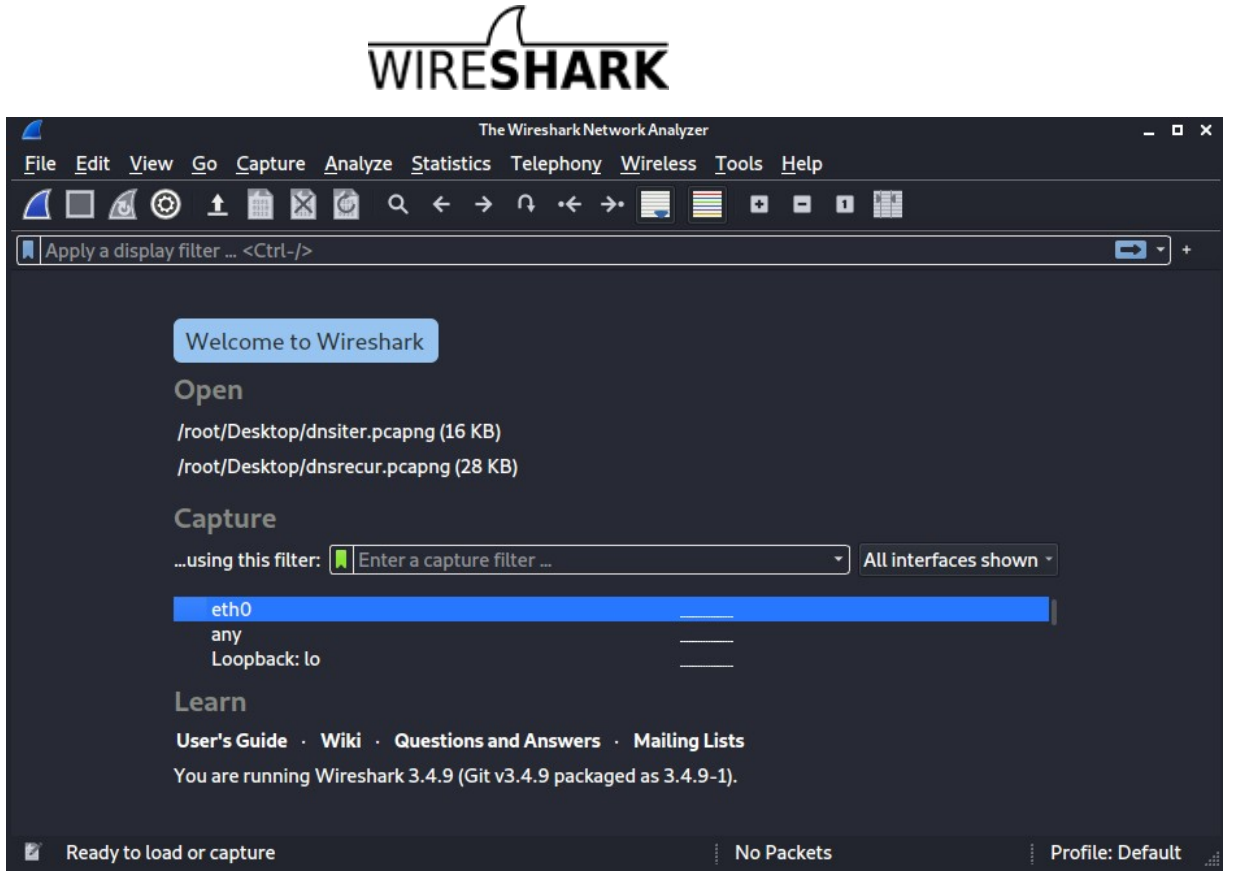
Windows Network Tools

- **netstat** – Sockets open in the machine



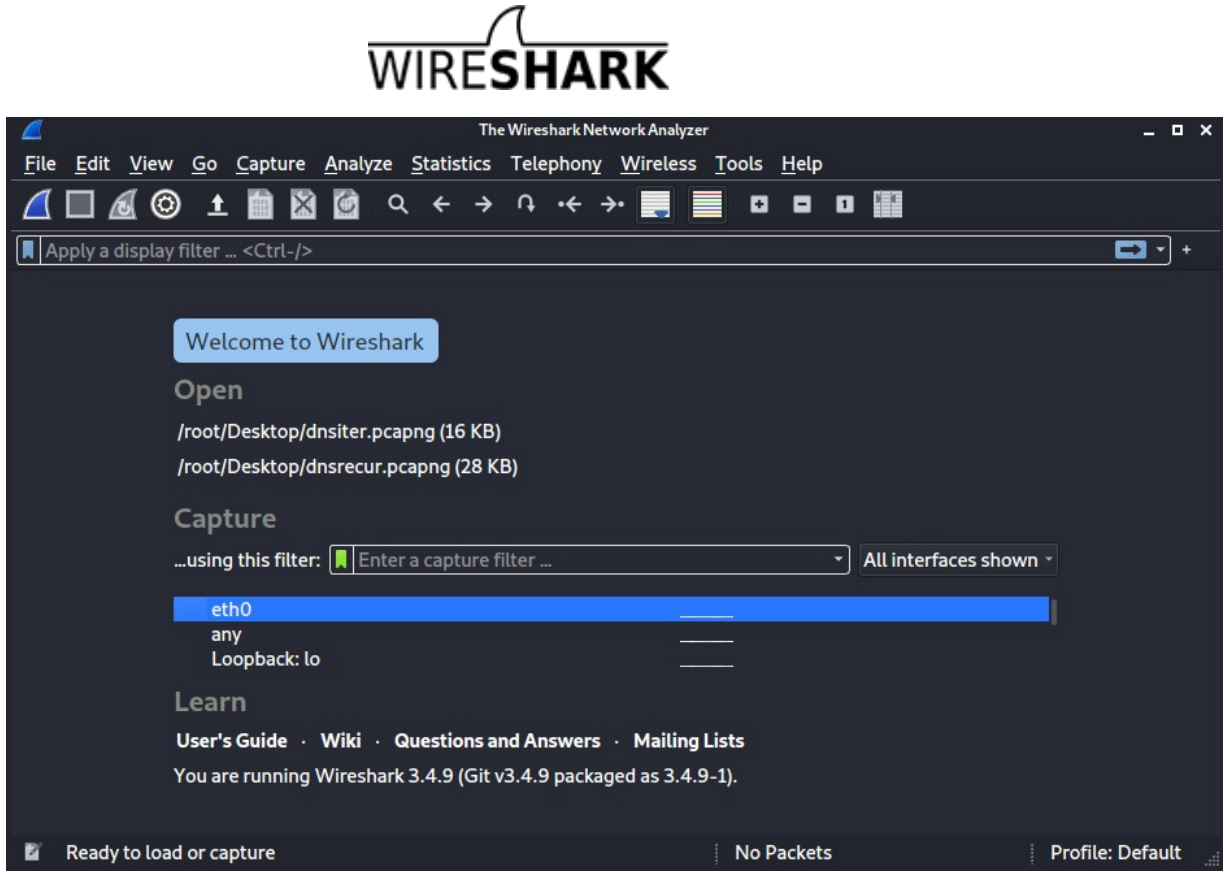
Packet Capturing

- Wireshark is a network protocol analyzer
- It can help capturing and inspecting traffic
- It is very useful to study network protocols and to detect network problems/attacks
- Start by opening it using the **wireshark** command



Packet Capturing

- Select the network interface that you desire to capture traffic
  - In the lab “eth0”
- Then press the fin on the top left of the application
  - This will put the interface in promiscuous mode - Accepting all traffic, even the one that is not intended for it.

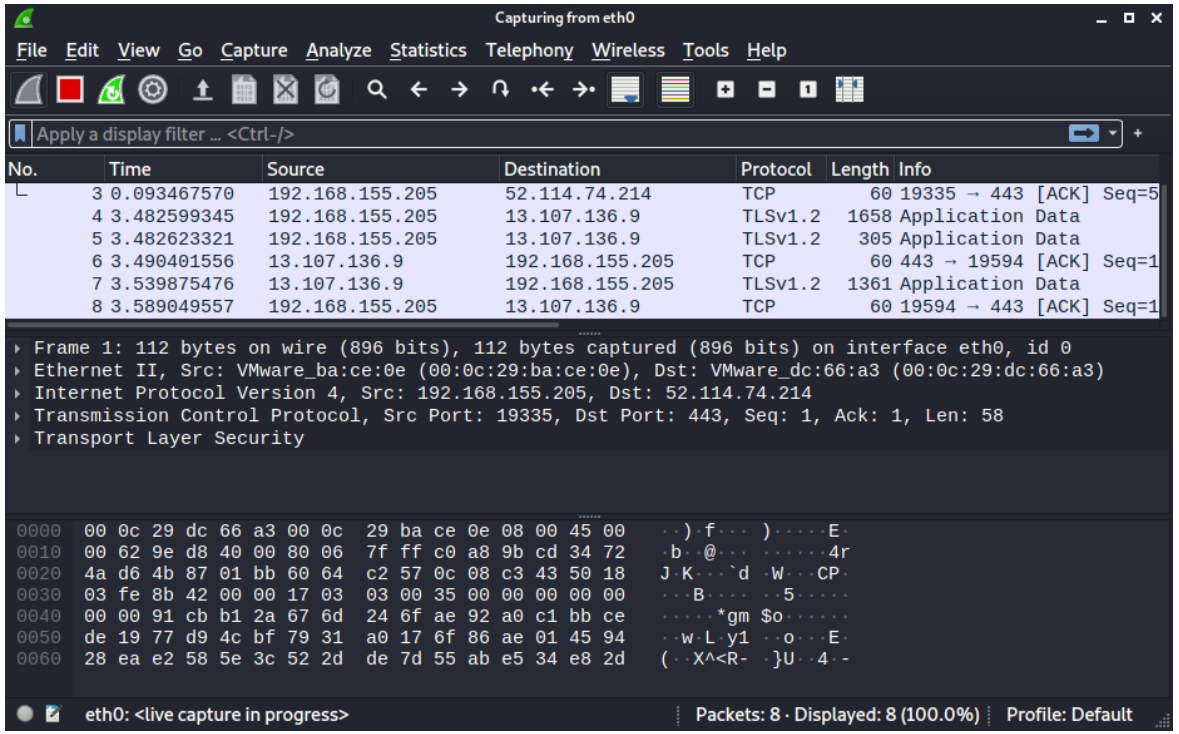


Wireshark intro

Packets Captured

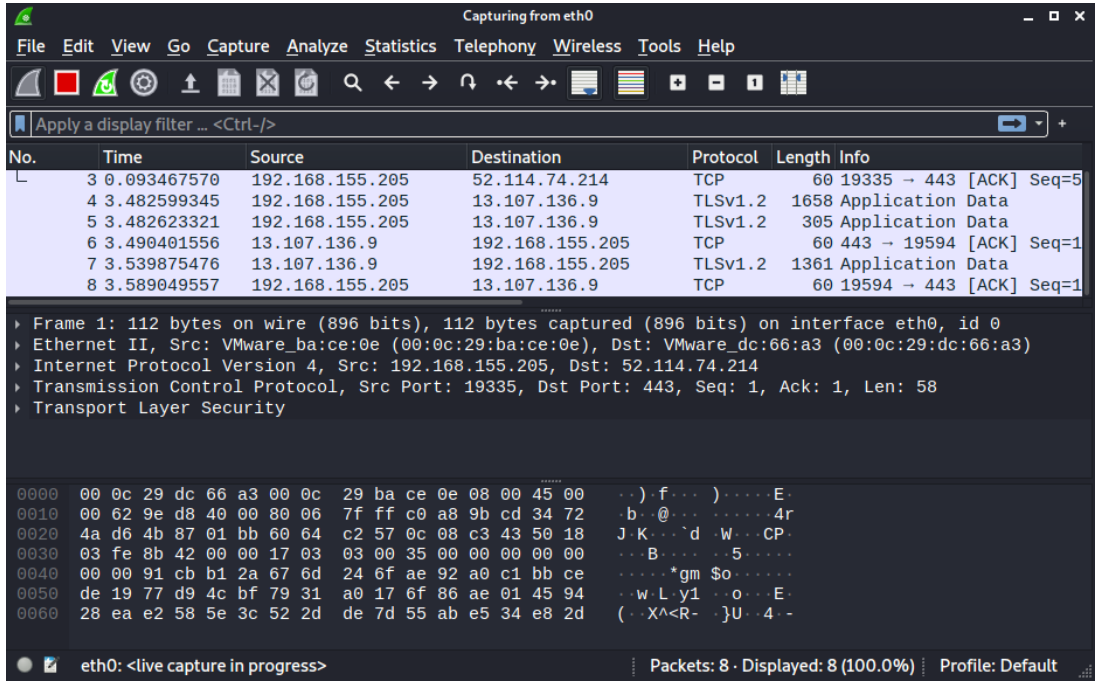
Packets Decoded

Raw View



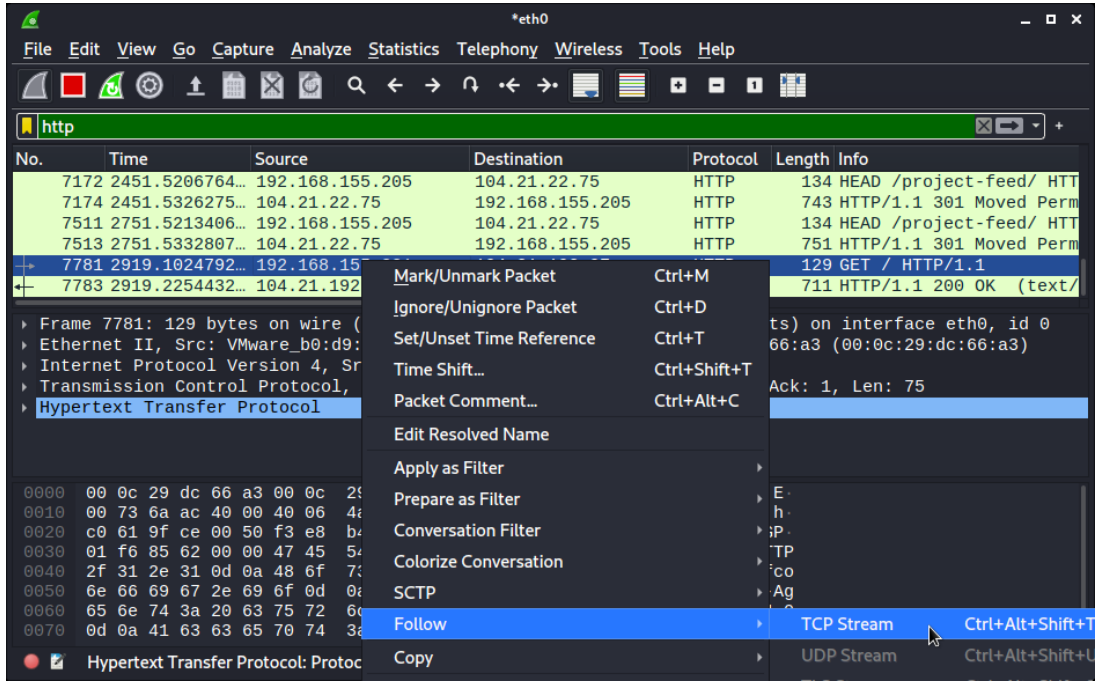
Packet Capturing

- In this view we start to see the traffic flowing
- We can select a packet for inspection
- We can also set a filter to filter out packets in the “Apply a display filter” bar



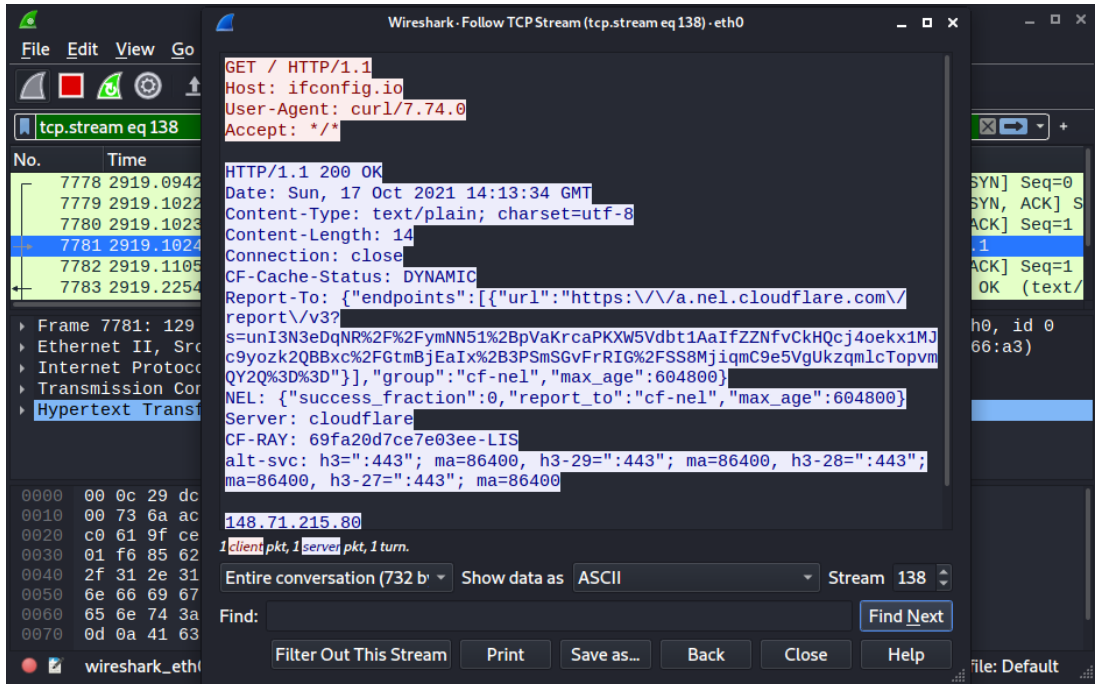
Wireshark intro

- Context Menu in each packet:
  - You can mark packets;
  - Type comments (useful for protocol debugging);
  - Use them as filters;
  - Follow the packet stream (for a complete transcript of messages during that session)



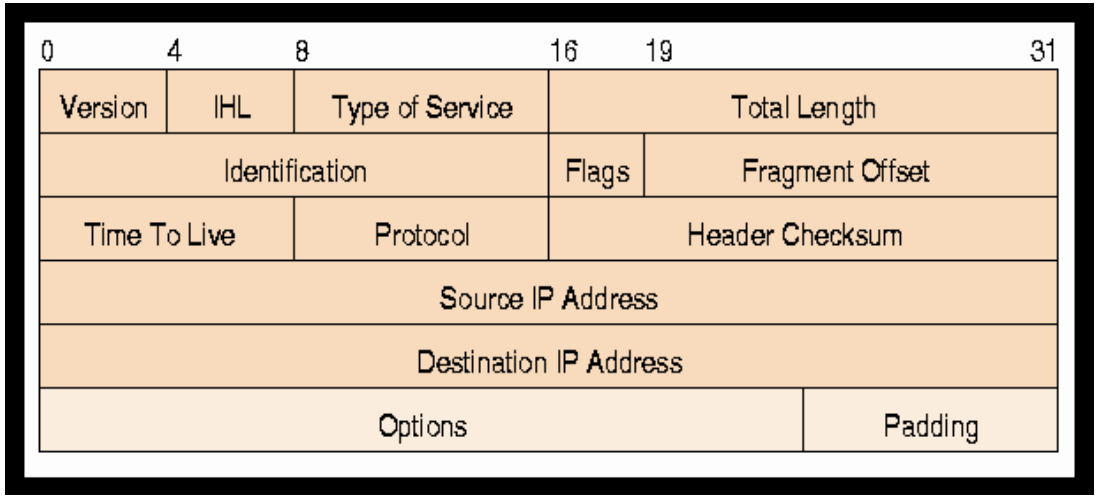
Wireshark intro

- Following the TCP/UDP stream will reassemble the entire conversation between the client and the server



Reading network data structures

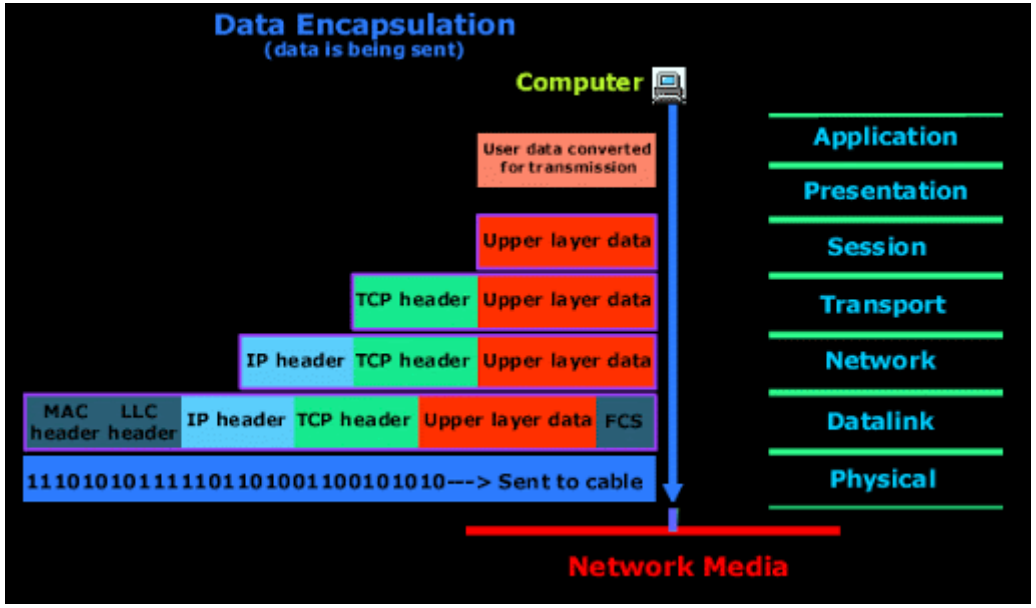
- Every packet has a specific structure, a set of values particular to that packet
- Giving a specific offset and a length of the field we can extract the necessary information
- Some of these structures are well known, such as the basic IP packet (on the right), and tools can automatically decode them.



https://www.freesoft.org/CIE/Course/Section3/7.htm

Packet

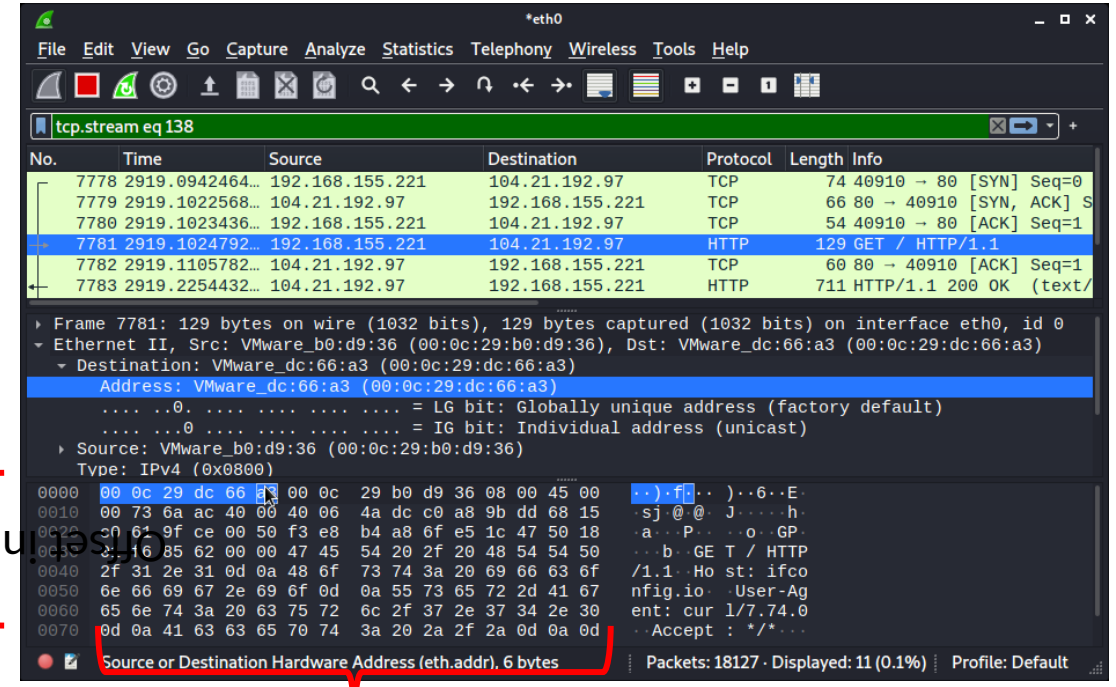
- The packet is an aggregation of layers each one independent of each other (so it is easy to swap)
- Each layer has a structure of information
- Before sending all these layers, they need to be encapsulated
- Receiving, the information is decapsulated



Ref: http://www.firewall.cx/networking-topics/the-osi-model/179-osi-data-encapsulation.html

Packet in Wireshark

- By hovering the mouse in the “Raw view” Wireshark will highlight certain bytes;
- By clicking the dissected view will show what values they represent and their meaning



Filters

- By protocol (**http**; **dns**; **icmp**; **smtp**; **smb**; **rdp**; **sstp** …)
- By field in protocol (**http.host==dei.isep.ipp.pt** ; **dns.a==1.1.1.1**)
- Boolean operations are also supported (**http.host==dei.isep.ipp.pt && http.request.method==GET**)
- Offsets are also possible (**tcp[0xd]&2**, SYN flag active in the TCP section offset **0xD**)
- More information: https://www.wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html