REQUIRED NETWORK KNOWLEDGE FOR CYBER SECURITY

NOTE: I didn't do e. Anatomy of Wi-Fi Frames 75, f. Wireshark Filters for Wi-Fi Frames 78, 3. Network Analysis 38 and 4. Linux Firewalls 58 Parts of the OTW. I'll do it later with the Wireshark pdf.

REFERENCE MODELS

Network Topologies

Bus Topology

To avoid signal reflection, a physical bus topology requires **each end** of network to be **terminated** with one end also being **grounded**. Loose or missing terminators from a bus network disrupts data transmission.

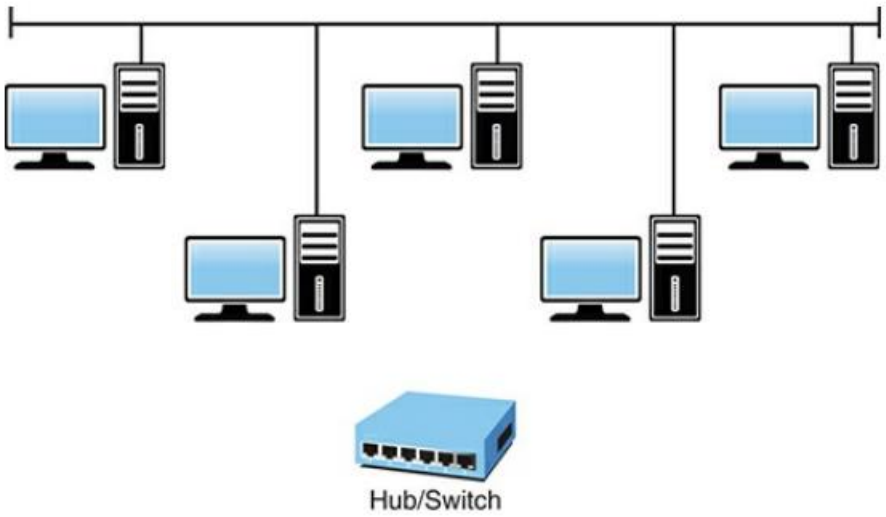
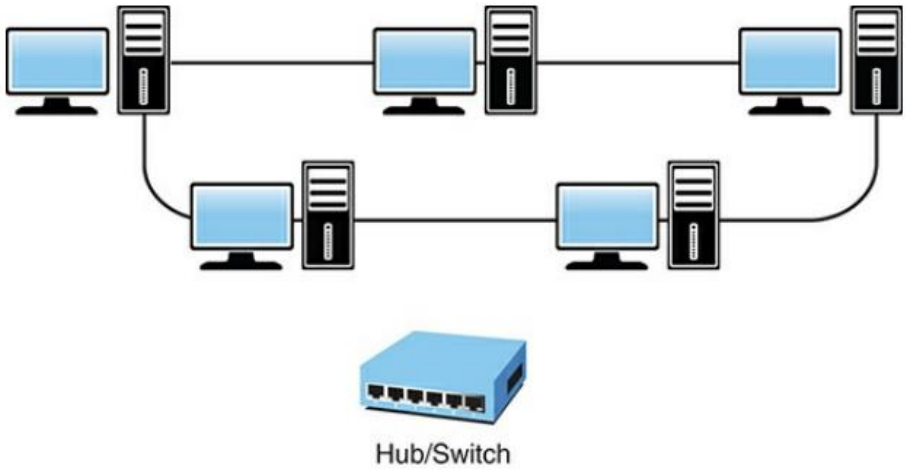


FIGURE 1.1 **Physical bus topology**

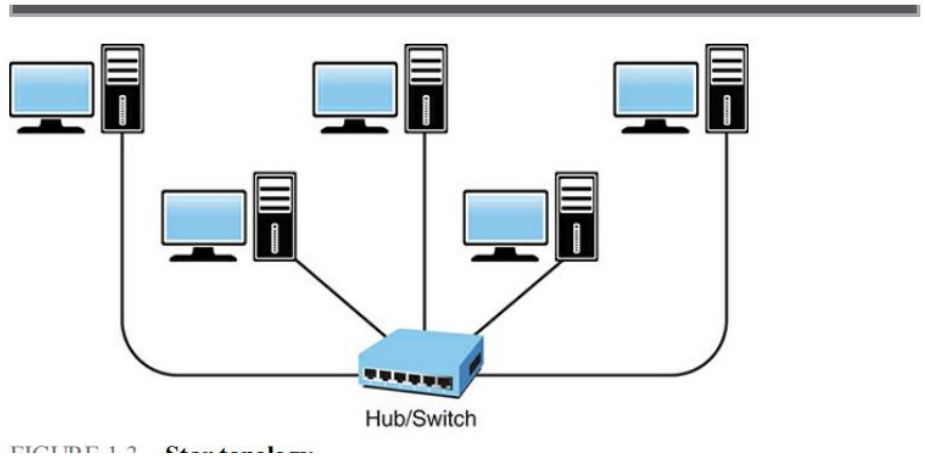
Ring Topology

If single cable fails all fail, but failures are easy to find. Ring networks are most commonly wired in a star configuration. They are easy to install, but expansions can cause disruption.



Star Topology

Star topologies are easiest to expand. As imaginable cable failures don't disrupt the entire network and troubleshooting is easier. But it requires more cable, and it requires additional networking equipment to create a network layout.



Wired Mesh

Each of the computers are connected to each other in this network. So, it provides a better fault tolerance than other topologies. And expansion in network don't cause disruption. But it is complicate to implement.

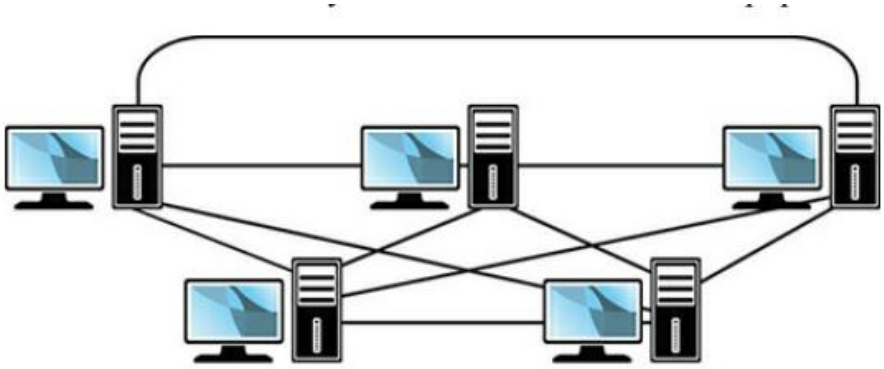


FIGURE 1.4 Mesh network

Wireless Topologies

Wireless networks typically are implemented using one of the 3.

->The infrastructure, or managed, wireless topology

->The ad hoc, or unmanaged, wireless topology

->The mesh wireless topology

Infrastructure Wireless Topology

It's generally used to extend a wired LAN to include wireless devices. Wireless devices communicate with the wired LAN through a base station known as access point (AP). WAP and APs are same things.

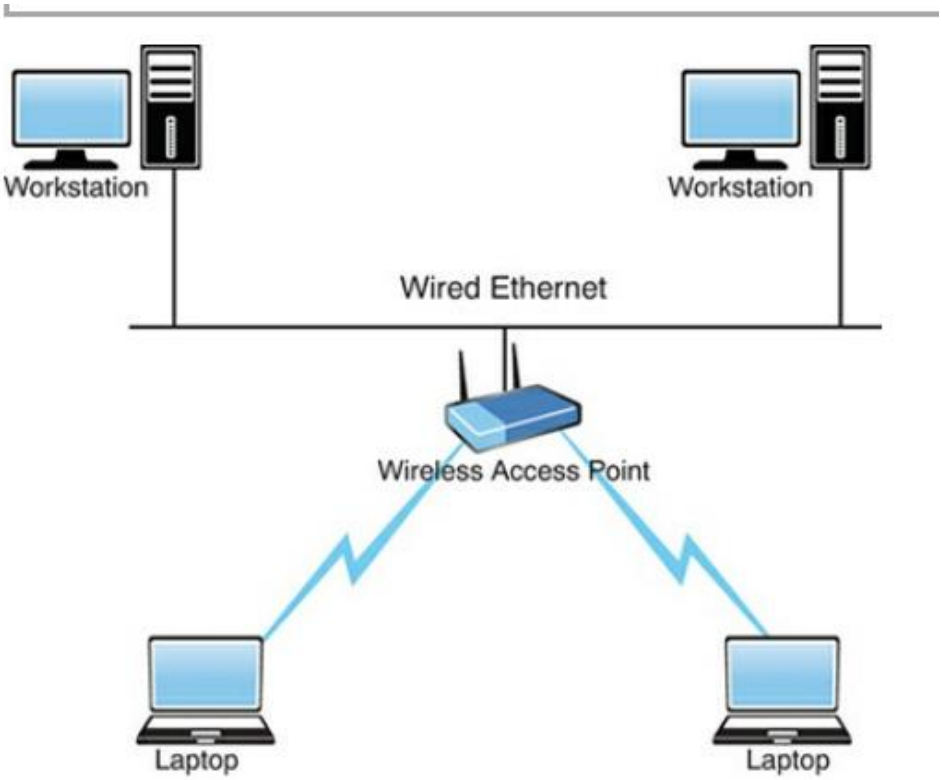


FIGURE 1.6 **Infrastructure wireless topology**

An access point (AP) is a networking device that allows wireless-enabled devices to connect to a wired network. It acts as a central hub for communication in a wireless local area network (WLAN) and bridges the gap between wireless devices (such as laptops, smartphones, and tablets) and the wired network infrastructure.

Ad Hoc Wireless Topology

Devices communicate with each other without using an access point. This is peer-to-peer networking. It is used to share files and resources among a small number of systems. Connecting mobile devices together or to a printer using Bluetooth is an example of an ad-hoc network.

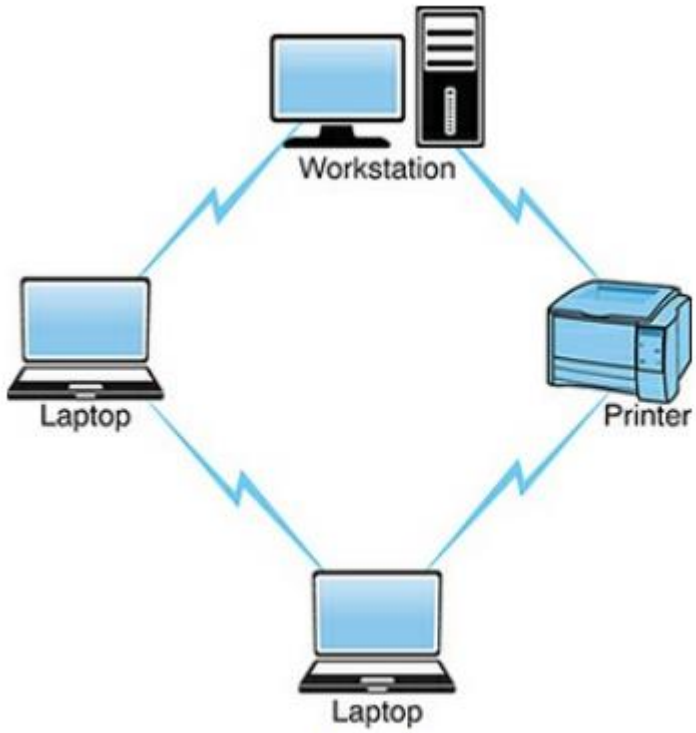


FIGURE 1.5 **Ad hoc wireless topology**

Wireless Mesh Topology

It is mesh. All devices connect to all devices.

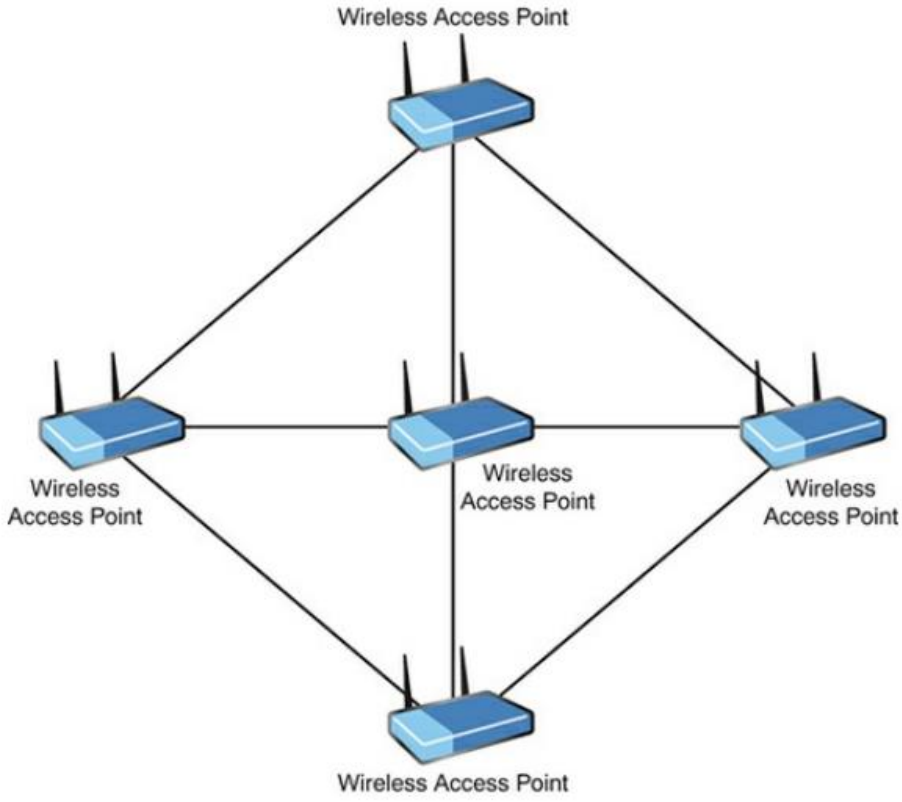


FIGURE 1.7 **A wireless mesh topology**

Hybrid Topologies

If you use more than one topology on a network than it is a hybrid topology.

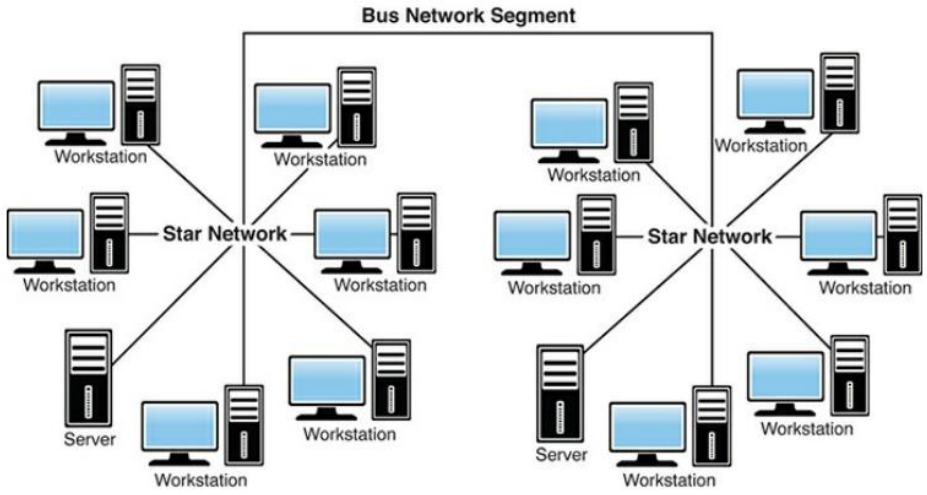


FIGURE 1.8 **A star bus topology**

LANs

Local Area Network is a data network that is restricted to a single geo-area, relatively small area such as office building or school.

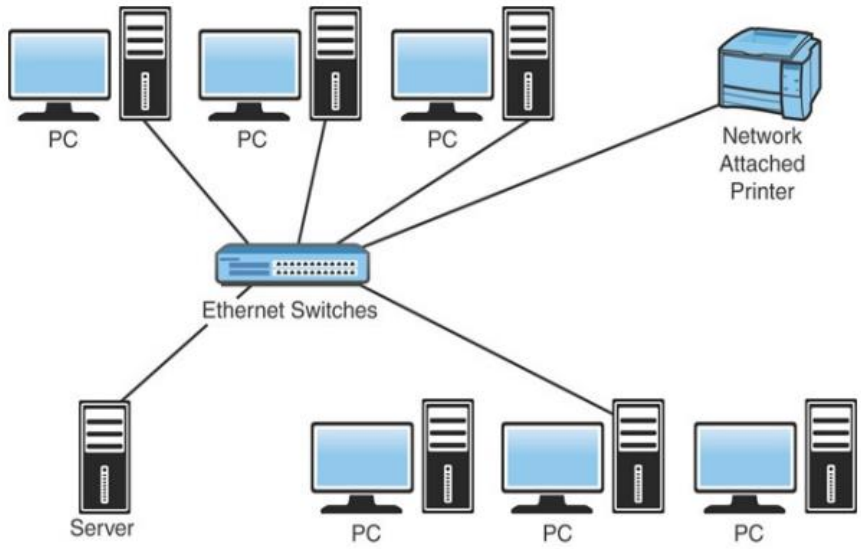


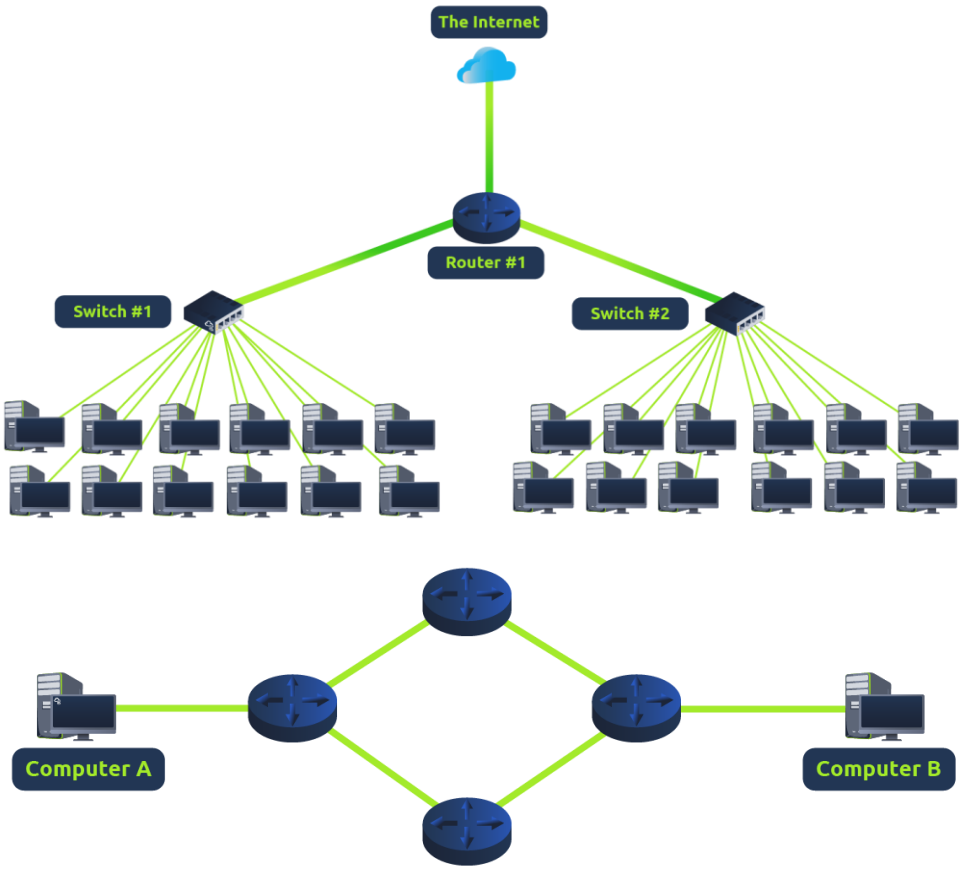
FIGURE 1.9 **A local-area network**

What is a Switch?

Switches are connectivity points for ethernet network.

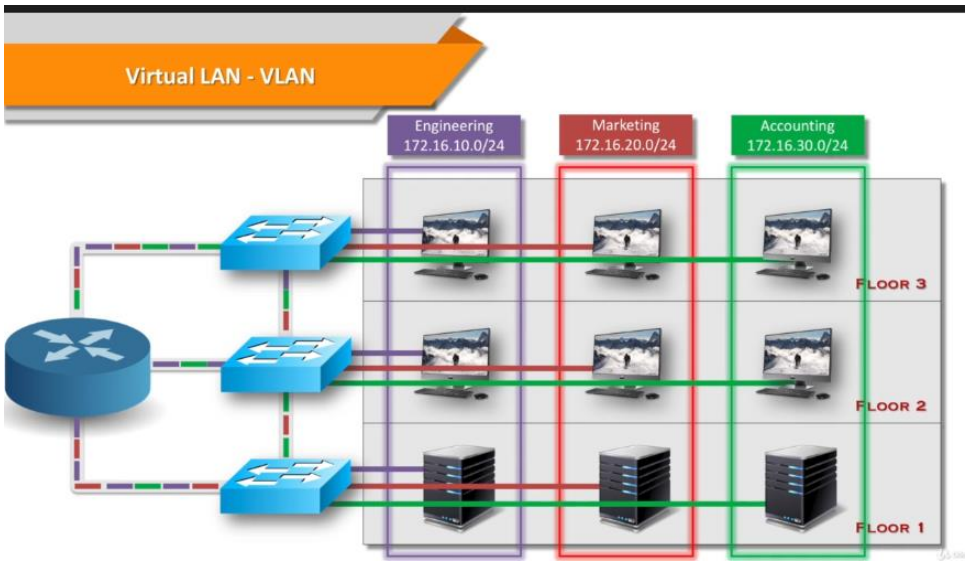
What is a Router?

Routers are devices that direct data between networks.



VLANs

VLAN or Virtual LAN. Lan is Local Area Network. Group of computers that are sharing a common communications line. **VLAN**s basically **segmentate a network**.



---> Computers and other devices use LAN to share resources (printers and etc)

---> VLAN is partitioned and isolated in data link layer.

---> In a VLAN the computers, servers and other network devices are logically connected regardless of their physical locations.

VLANs enable you to create multiple broadcast domains on a single switch. In essence, this is the same as creating separate network for each VLAN.

BUT WHY USE VLAN?

Security = We separate sensitive data from the rest.

Cost Reduction = Less need for expensive network upgrades.

Higher Performance = Reduce the number of router hops and increase the apparent bandwidth

Simpler Management = For specific employees.

WLANs (Wireless Local Area Network or IEEE 802.11)

---> Wireless Networks **connects nodes** using radio communication.

---> By doing this we can avoid expensive cable connections.

---> ---> Wireless Local Area Networks (WLANs) links devices over a short distance using an access point.

---> ---> ---> MOST Significant Wireless Network Attacks.

Denial Of Service:

Rogue Access Points: Setting up unauthorized access points.

Misconfiguration: Access points that are not configured appropriately.

Traffic capture / Intercept: Sniffing the network.

Crypto Attacks: Weak wireless encryption protocols.

Client-duping: Setting up a bogus access point.

---> ---> ---> ---> OTW Wi - Fi Networks (802.11)

---> Terminology

AP: This is the access point for the clients to connect to the Wi-Fi and get access to the internet.

PSK: Or Pre-Shared-Key. PSK's use WPA. It is used to authenticate to the AP. previously shared between the two parties using some secure channel before it needs to be used

SSID: AP name

ESSID: It is the same as SSID, but ESSID can be used for multiple APs in a wireless LAN.

BSSID: It is unique identifier for each AP. It is same as MAC address of the AP.

Modes: Wi-Fis are operating in 3 modes in general. Master, Managed and monitor. APs operate in master mode, wireless network interfaces operate in monitor mode.

Frequency: Either 2.4 or 5 GHZ

---> Security Protocols of Wi-Fi

WEP: This was the initial protocol at first, but due to the vulnerabilities with it (Due to the improper implementation of RC4 encryption), it is rarely used today.

WPA: This protocol also used RC4 encryption. But it added some additional features to make PSK cracking hard such as

1: Making the initialization vector longer from 48 to 128 bits.

2: TKIP, Which generates different keys for each client.

3: Message Integrity Check to make certain messages haven't been altered enroute.

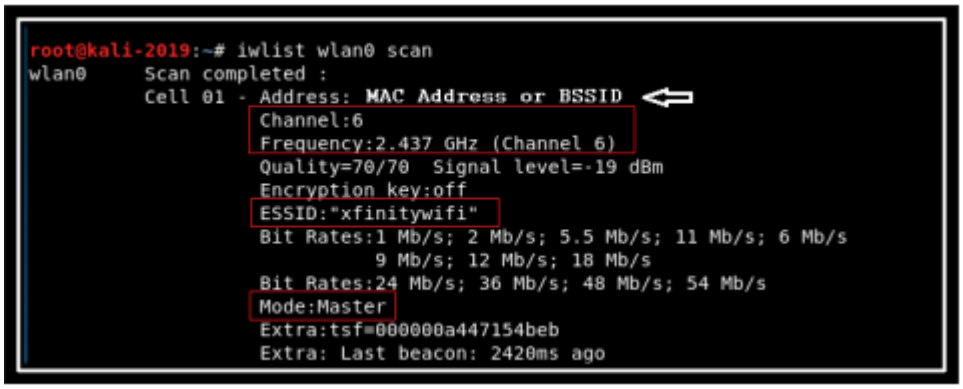
WPA2: Uses Cipher Block Chaining Message Authentication Protocol AKA CCMP. However, because CCMP is more processor intensive it requires a better hardware. CCMP protocol is based on AES encryption. WPA2 supports both Personal and Enterprise modes. When using the personal mode (PSK), the pre-shared key (password) is combined (AKA salted) with SSID to produce PMK (Pairwise Master Key) to make rainbow tables harder.

---> Commands

---> To see our wireless extension, we can use **kali> iwconfig**

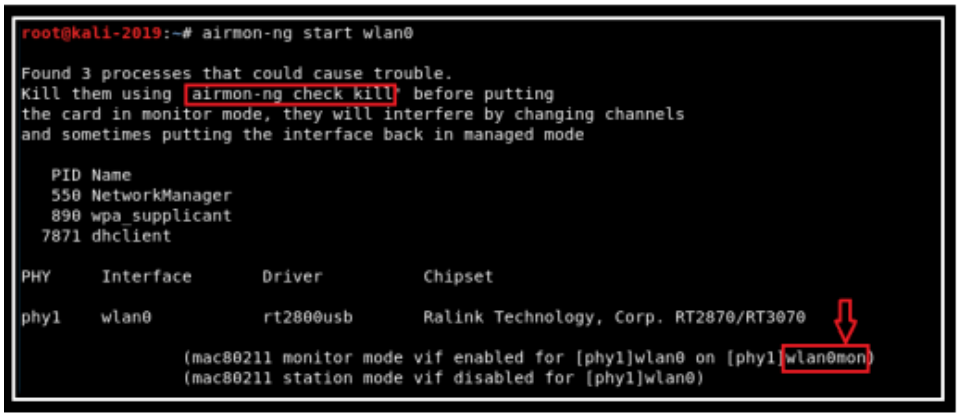
As we can see, our extension is wlan0.

---> To view all the Wi-Fi APs within range of your wireless network interface, you can enter "iwlist" command in Linux. (kali > iwlist)



---> Monitor Mode: (From ChatGPT) In monitor mode, the wireless device can passively listen to all wireless traffic on a specific Wi-Fi. But environmental factors are also important for the sniffing process. Since the device is not connecting to the network its ability to sniff is limited to the area covered by the Wi-Fi signal from the target network.

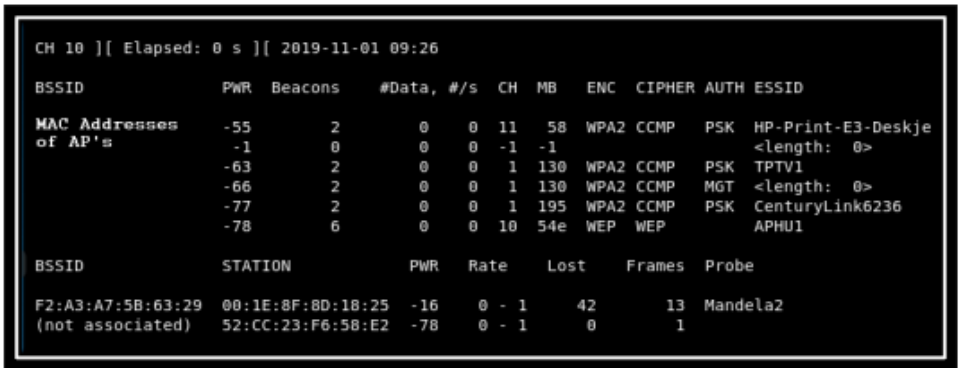
---> To enter monitor mode we enter: **kali> airmon-ng start wlan0**



And we can see that our NIC has entered monitor mode.

---> If there is a problem, we can deal with it by: **Kali> airmon-ng check kill**

---> And now we are ready to sniff the network: **kali> airdump-ng wlan0mon**



And now we can see all the APs with their critical info.

---> Attacking Wi-Fi APs

---> The security engineers taught to hide their SSIDs so only the people who know the SSID can be able to discover and connect to their Wi-Fi, however this is wrong. A hacker doesn't need to know someone's SSID to join, hacker needs to know MAC address. And this information is broadcasted over the air so the hacker can sniff it with tools such as airodump-ng or others to view the BSSIDs. As shown before.

---> The security engineers are taught to limit who can access their Wi-Fi AP by using MAC filtering to make sure only the approved MAC addresses can enter the network, However, a hacker can easily spoof it's MAC address by first listening to the network with:

kali > airodump-ng –c 11 –a –bssid <MAC> and after seeing the client's MAC address, hacker can change his/her MAC address with:

kali> ifconfig wlan0 down after this

kali> macchanger –m <Target_MAC> wlan0 and then

kali> ifconfig wlan0 up

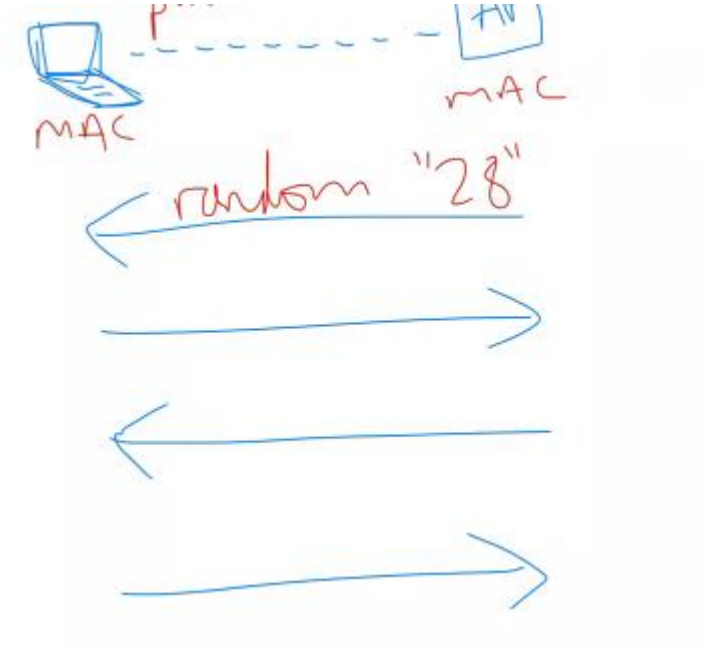
---> Attacking WPA2-PSK

---> What we do is basically we first sniff the network to get the hash, and then we apply wordlist such as hashcat to find a match.

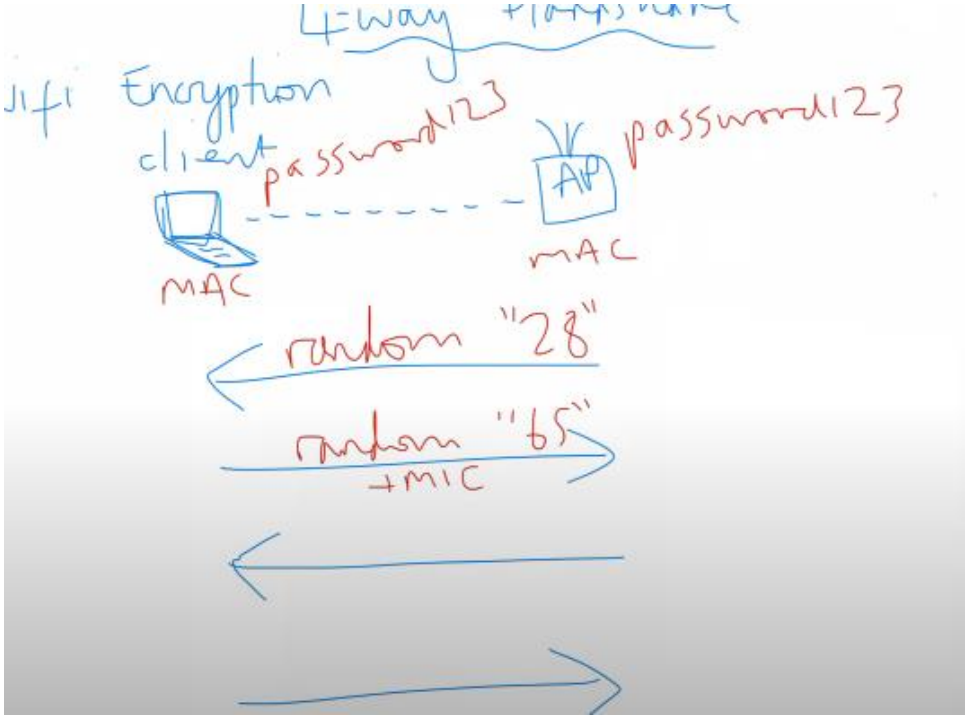
---> WPA2-PSK has four-way handshake, where the password hash is transmitted across the air between the client and the AP. Before continuing I want to get deep into 4-way handshake with Jennifer-Watson Youtube channel.

At first the computer and AP only know each other's MAC address. And have their Wi-Fi passwords but they don't want to send that information to each other for safety reasons. So, four-way handshake is a mechanism that makes sure any data sent between client and AP is encrypted.

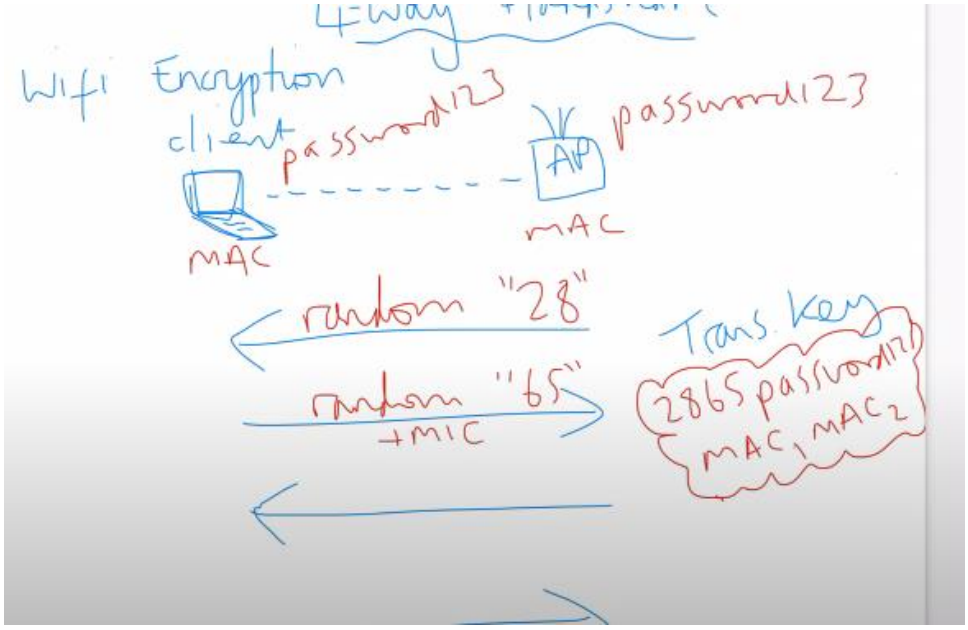
1.) AP sends client a random number.



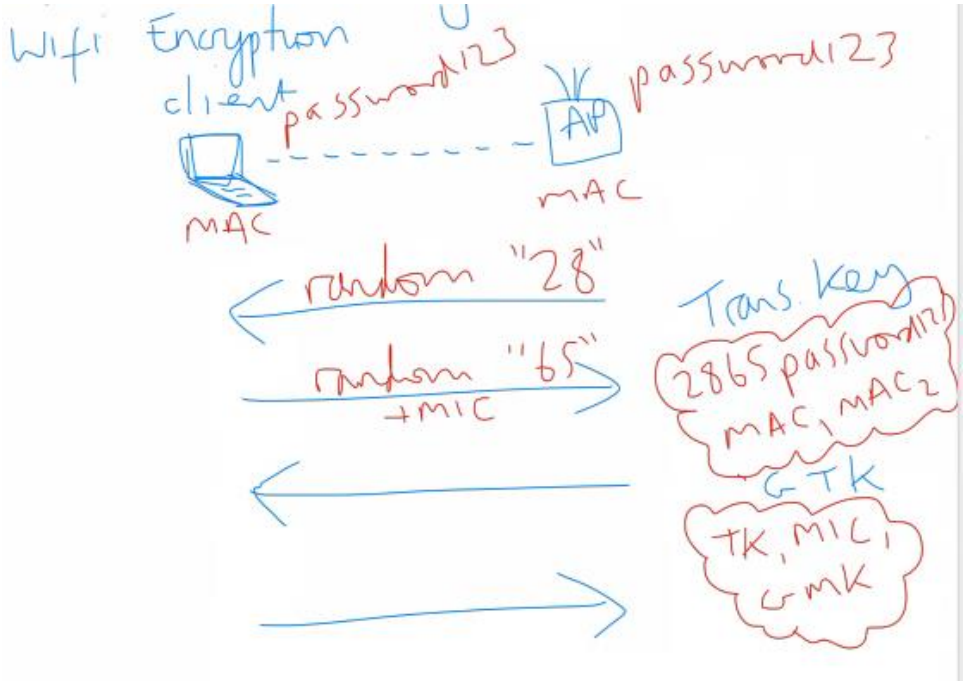
2.) Client sends another random number to the AP with the MIC (Message Integrity Code which is clients id to prove it is what it says it is.)



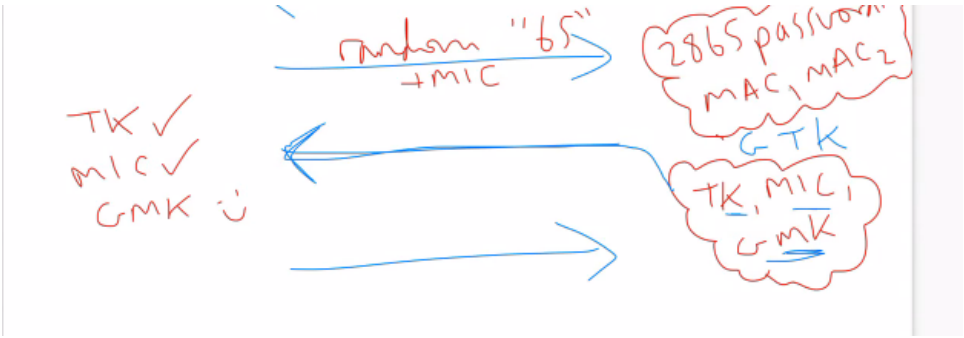
3.) Now both devices have their own MAC address, others MAC address, password, and random numbers. And they put that info together, independently, to a thing called transient key.



Access point now needs to communicate back to the client. So, it takes trans key with Message integrity code with Group Master Key (GMK). GMK is important for it determines how data is encrypted. And TK (trans key), MIC and GMK are put together to GTK.



4.) They send each other message confirming the client's identity.



---> So now let's get back to the OTW.

---> First, we need to put our NICs into the monitor mode.

kali> airmon-ng start wlan0

---> Then we start collecting information and packages.

kali> airodump-ng wlan0mon

```
CH 10 ][ Elapsed: 0 s ][ 2019-11-01 09:26

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

MAC Addresses      -55        2        0    0  11   58  WPA2 CCMP   PSK  HP-Print-E3-Deskje
of AP's             -1        0        0    0  -1   -1                   <length:  0>
                   -63        2        0    0   1  130  WPA2 CCMP   PSK  TPTV1
                   -66        2        0    0   1  130  WPA2 CCMP   MGT  <length:  0>
                   -77        2        0    0   1  195  WPA2 CCMP   PSK  CenturyLink6236
                   -78        6        0    0  10  54e  WEP  WEP         APHU1

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

F2:A3:A7:5B:63:29  00:1E:8F:8D:18:25  -16    0 - 1     42        13  Mandela2
(not associated)   52:CC:23:F6:58:E2  -78    0 - 1      0         1
```

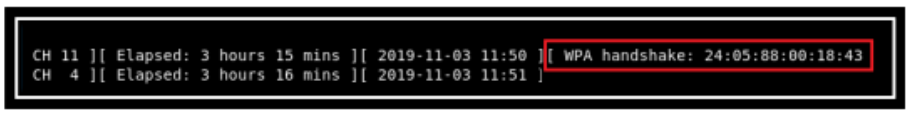
---> We'll focus on packet capture on a single AP on a single channel. We can do that by entering the following:

kali > airodump-ng –bssid -c --write wlan0mon

---> We can use deauth attacks to get handshakes. With:

kali > aireplay-ng –deauth 100 –a AA:BB:CC:DD:EE:FF wlan0mon

---> And now we captured the handshake.



What we need to do know is to crack the hash with programs such as hashcat to brute force it.

This can be slow and tedious, so a good wordlist is a must. For that you can try:

kali > hashcat –m 16800 HackersAriseCrack-01.cap /root/top10000passwords.txt

If you are unsuccessful with this you can use ceWL, cup, crunch to create a custom wordlist.

IMPORTANT NOTE: This isn't finished yet but I know the rest of it so I may fulfill this part later on. Page 85.

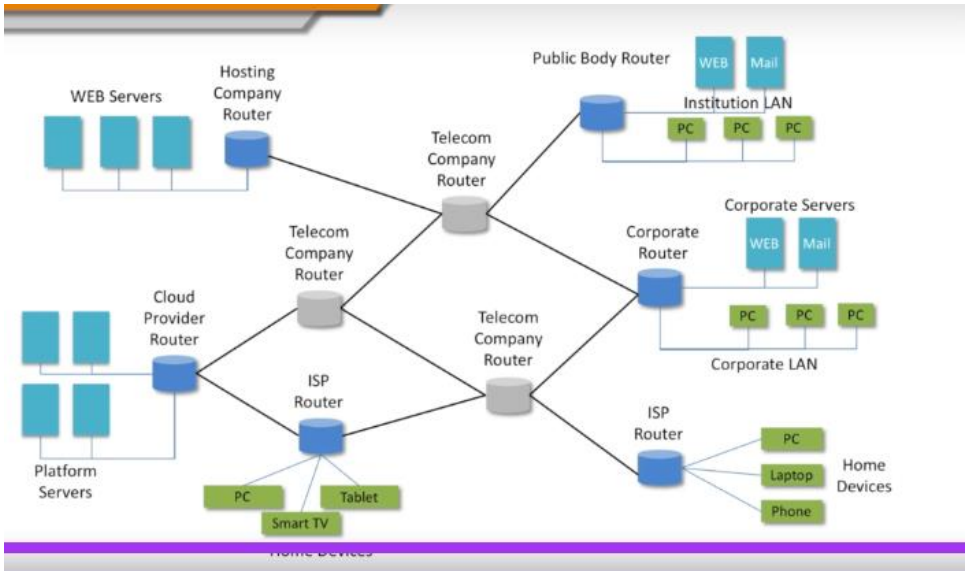
What Is Protocol?

A protocol is set of rules that acts as a language. They provide a set of rules and standardization.

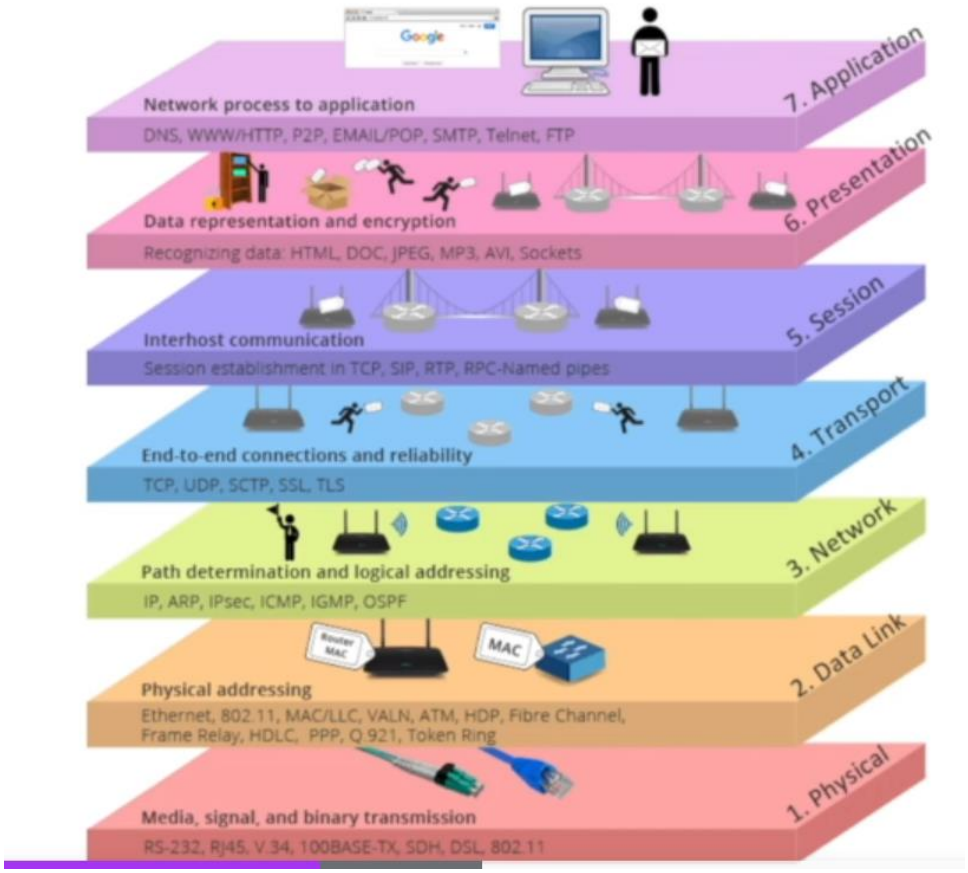
IP = Internet Protocol, describes how packets move. IP protocol **standardizes** how computers forward or rout their packets based on their IP addresses.

IP routing= Forwards IP packets from source to destination

IP address = Unique address that identifies machines.



OSI Reference Model



---> Mnemonic for OSI: **A**nxious **P**ale **S**hakespeare **T**reated **N**ervous **D**runk **P**atiently

---> **Layer 7: Application Layer** = Communication partners are identified. Is there someone to talk to? (Ex: HTTTP SMTP) This layer is not applications itself.

---> **Layer 6: Presentation Layer** = Part of OS. Convert incoming and outgoing data. Like from clear to ascii, MP3and vice versa.

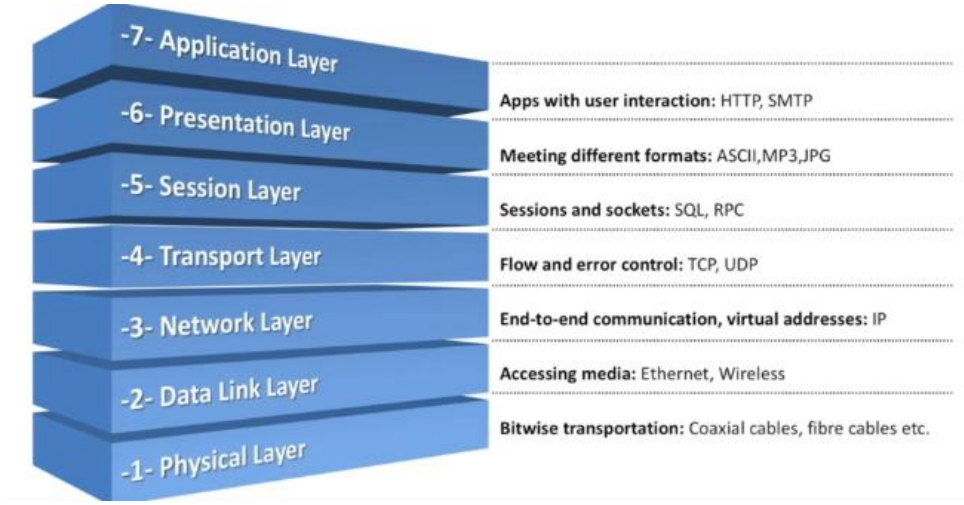
---> **Layer 5: Session Layer** = Responsible for managing and controlling synchronization of data. Sets up coordinates and terminates connections (SQL, RPC, NetBIOS, NFS, SMB are here)

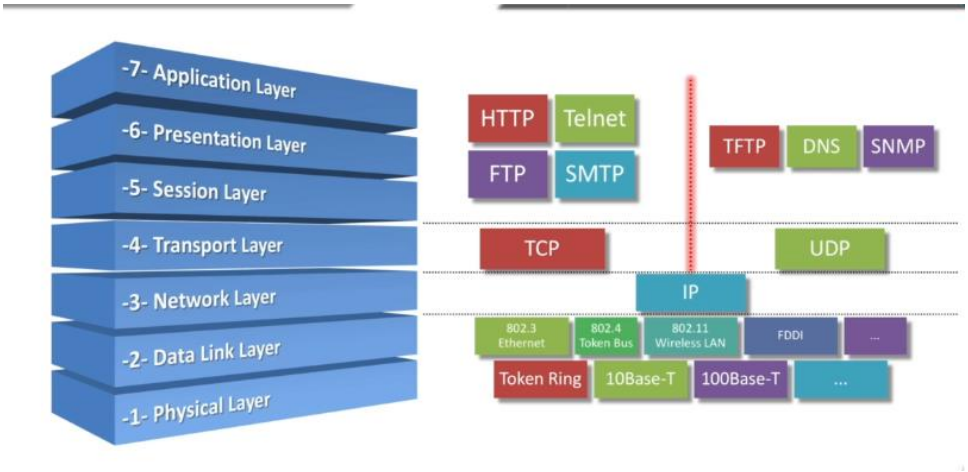
---> **Layer 4: Transport Layer** = Checks the packetization of data and delivery of packets. It does **error checking**, **service addressing**, (it ensures that data is passed to the right service at the upper layers of the OSI), **segmentation** (Brokes down the data), **buffering** (Stores data temporarily, waits destination device to become available) (TCP and UDP are here)

---> **Layer 3: Network Layer** = Addressing and **routing** of the data. Sending it to the right direction and receiving data (IP, ARP, RARP, ICMP are here)

---> **Layer 2: Data Link Layer** = Sets up links in physical networks. it has 2 sublayers. Ethernet and wireless. The data link is mainly responsible for getting data to the physical layer so that it can be transferred. Also responsible for error detection, error correction and hardware addressing. MAC address and logical link control and present the data in a format suitable for transmission

---> **Layer 1: Physical Layer** = Cables and stuff. Hardware and topology. USB, Ethernet are also here. Physical layers include the topologies to be used in the network as well.





Device	OSI Layer
Hub	Physical (Layer 1)
Wireless bridge	Data link (Layer 2)
Switch	Data link (Layer 2) or network (Layer 3)
Router	Network (Layer 3)
NIC	Data link (Layer 2)
Access point (AP)	Data link (Layer 2)

---> You wrote a URL and pressed it. The application layer sends the DNS query to the transport layer. (DNS query is a process of a network device to get an IP address from URL).

Client computer sends a DNS query to its ISPs. DNS server looks at DNS query to answer and if it can answer it answers it. Application, presentation and session layer is passed when this is done.

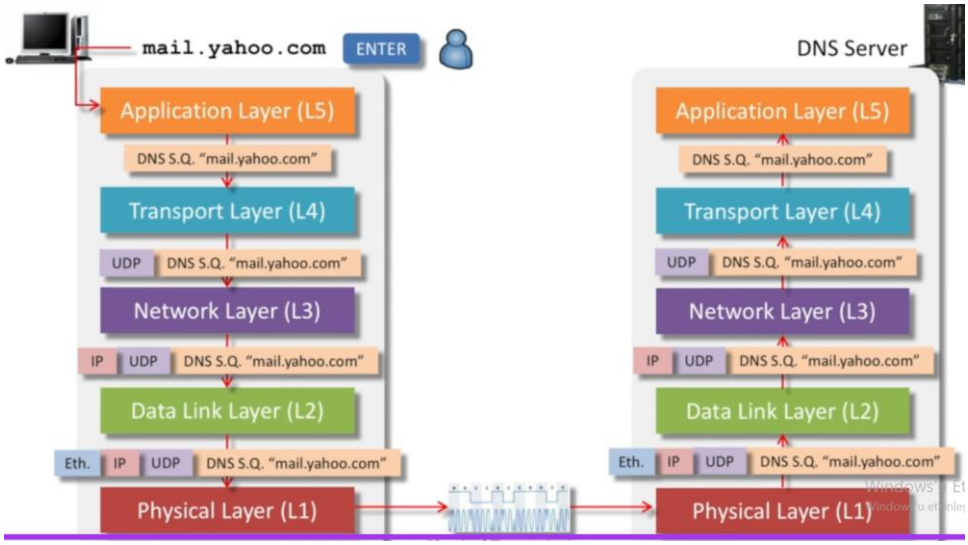
---> Let's assume DNS query is using UDP right now.

---> Network layer adds ip address of the source and destination device.

---> Data link layer makes a new data unit called the frame. By adding, the layer 2 frame header (Ethernet header here)

---> Physical layer turns data into binaries.

---> Just the same way it came to the physical layer, it is decapsulated

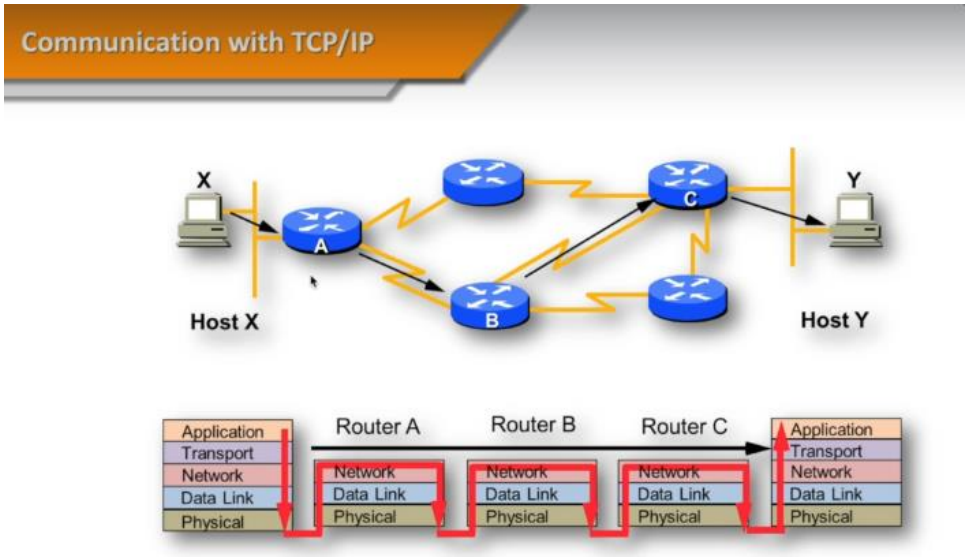


OSI vs TCP/IP

---> TCP/IP is not the same with OSI but most of the things are the same. TCP/IP is real life while OSI is ideal.

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	

---> TCP/IP doesn't make any references to the link layers below IP. But OSI does.



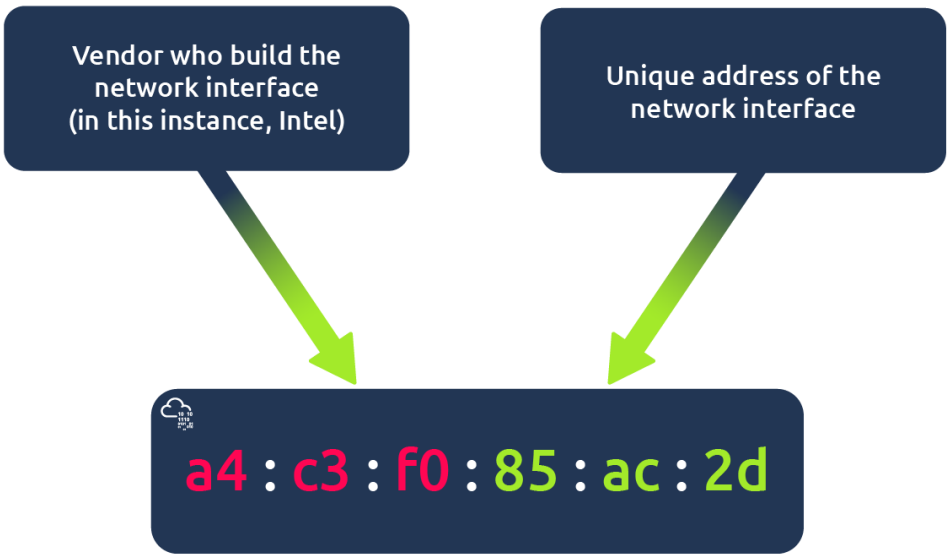
DATA LINK LAYER (LAYER 2)

---> Data link is responsible for encoding bits into packets before transmission. And it is also responsible for decoding packets back into bits at the destination.

---> It is also **responsible** for **logical link control**, **media access**, **hardware addressing**, **error detection**.

---> In every network enabled computer there is NIC (Network Interface Card). And inside NIC there is a MAC address which is burned into NIC. **MAC** Layer and Logical Link Control (**LLC**) Layer are also in data link layer. **Wi-Fi** operates in Layer 1 and 2

MAC



---> MAC is burnt into computer.

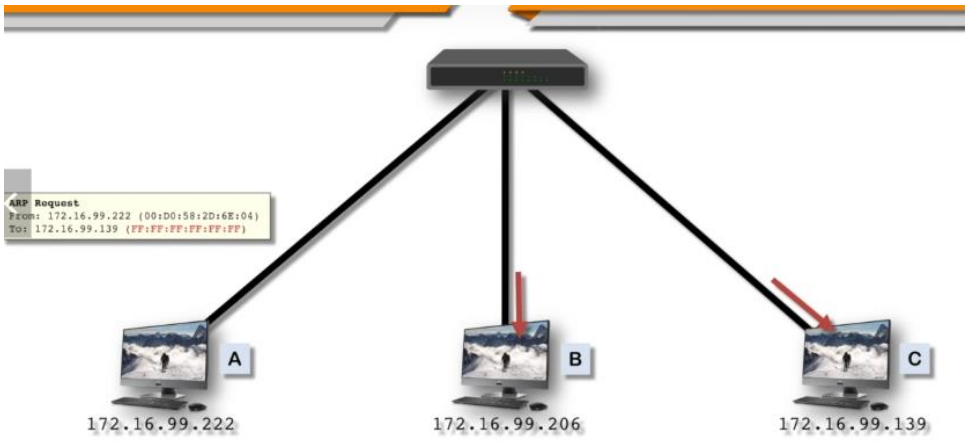
Ethernet

---> Multiple computers can send data at the same time.

---> Collision handling: Carrier Sense Multiple Access - Collision Detection (CSMA/CD)

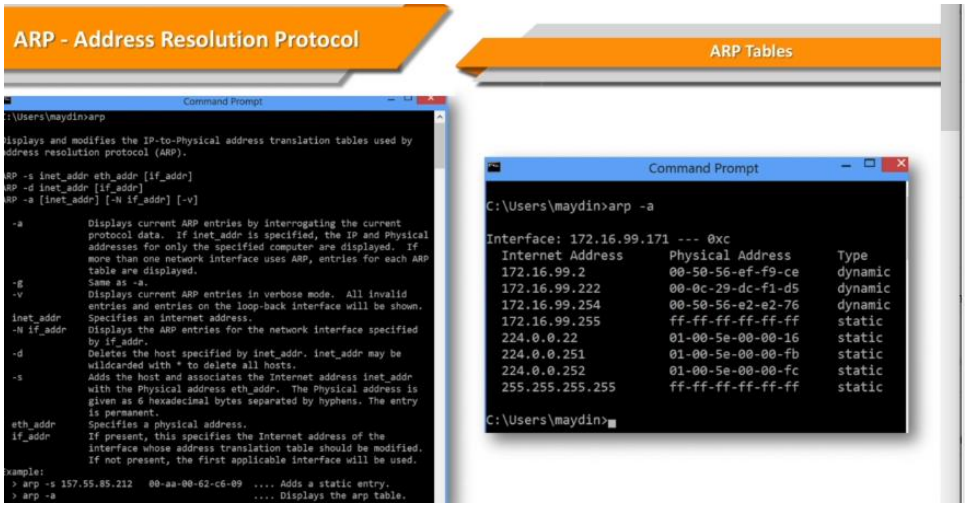
ARP: Mechanism, Arp Tables and ARP Packets

ARP (Address Resolution Protocol): Used for mapping a network address to a physical address. (Translates **IP to MAC**)



---> Computer A wants to talk to C. So, A broadcasts a message asking for C's address. Since it is a broadcast, it also goes to the B but since B knows that it is not C it doesn't responds. Only C responds.

---> Since the computers don't want to broadcast all the time, they are holding an **ARP table**. The ARP table changes in every 15 minutes or so. The ARP tables holds the addresses of other computers.

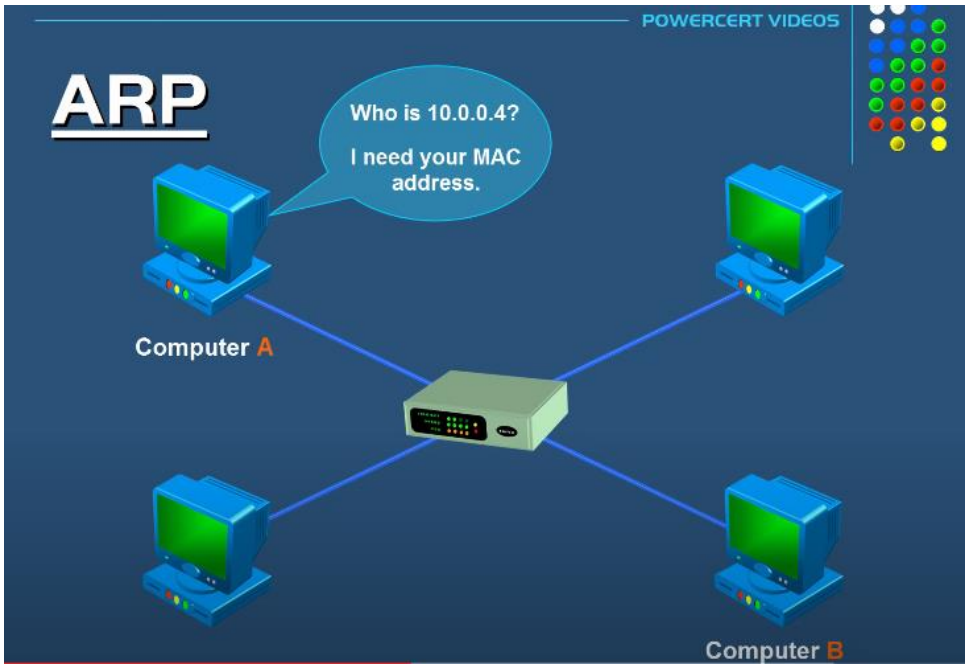


---> With arp -a you can see and read arp tables. (in windows)

---> ---> ---> ---> PowerCert Animated Videos ARP Explained

---> Resolves IP address to MAC address. Devices need MAC address for communicating on a local area network.

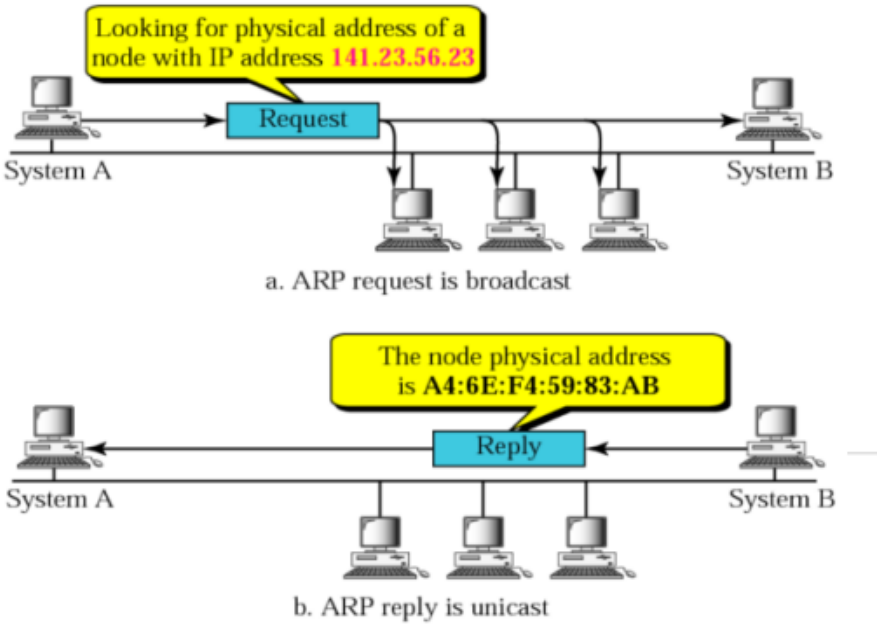
---> Lets say 2 computers want to communicate. Let them be computer A and B. Computer A already knows the IP of computer B but needs to know it's MAC address to communicate. It checks its ARP cache first.



---> After that it broadcasts a massage to check which computer has the MAC address of that IP address. And the requested computer returns its MAC address. And this is stored to the ARP cache.

---> ---> ---> ---> OTW ARP

---> ARP protocol can be used for assigning IP address as well.



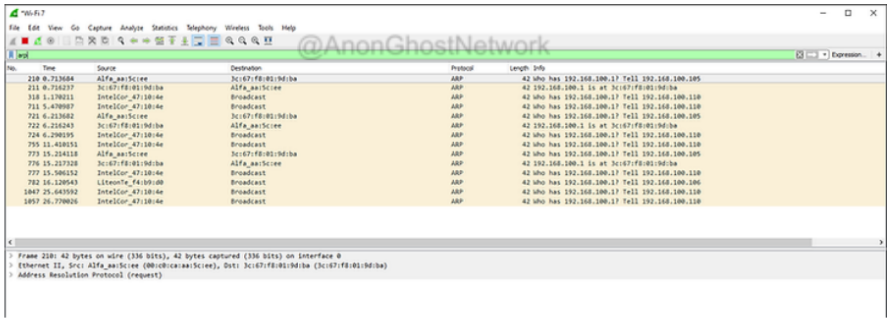
---> ARP basically works like this. When a new device enters the network

---> If the attacker understands ARP, they can conduct MITM attacks.

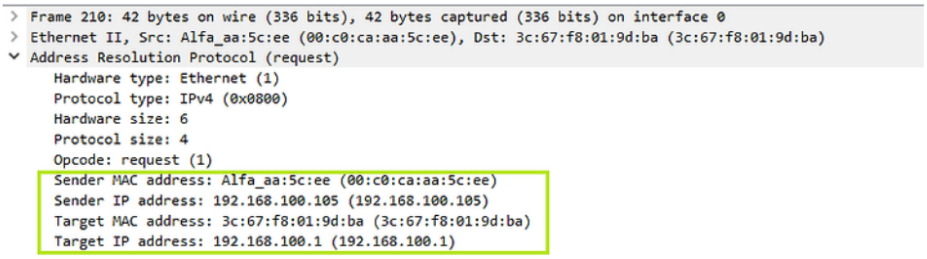
---> But how does it work? 2 computers on the same Local Area Network wants to communicate. To do that, in addition to IP they need MAC address as well. So, first computer firstly looks at its ARP Table to find target computer's MAC address. If found, no problem but if it can't find it then it sends broadcast to the each and every computer on the network and when the target computer returns "I have that IP Address". With it the computer1 knows the MAC address of target computer.

ARP Packets in Wireshark

We can view the arp packets in Wireshark by simply entering the word "arp" in the filter window like below.



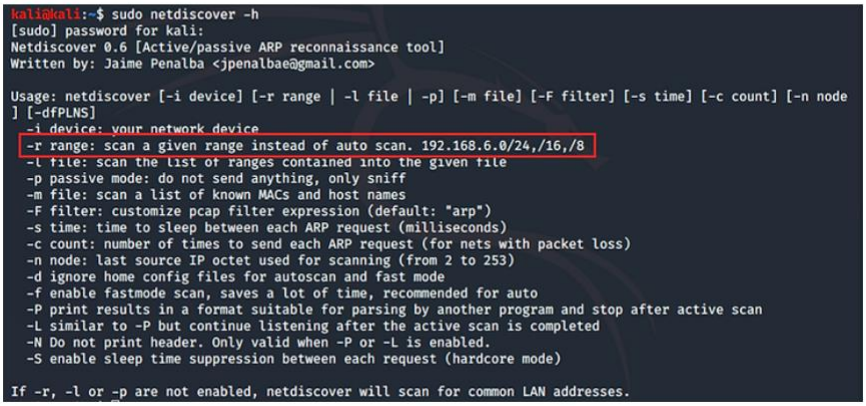
When we click on a single packet, we can dissect the packet. Expanding the Address Resolution Protocol field, we can see the Sender and Target IP and MAC addresses.



---> ARP doesn't have any authentication, so **bad actors can use this for discovering devices on the network**. This can be useful for hacking another device on the same network, or we can use this to hack more valuable target on the network such as a DB server.

---> One of the tools that can be used for discovery is **netdiscover**

kali > sudo netdiscover -h



As you can see above, we can use the -r option to scan a range of IP addresses on a network, such as;

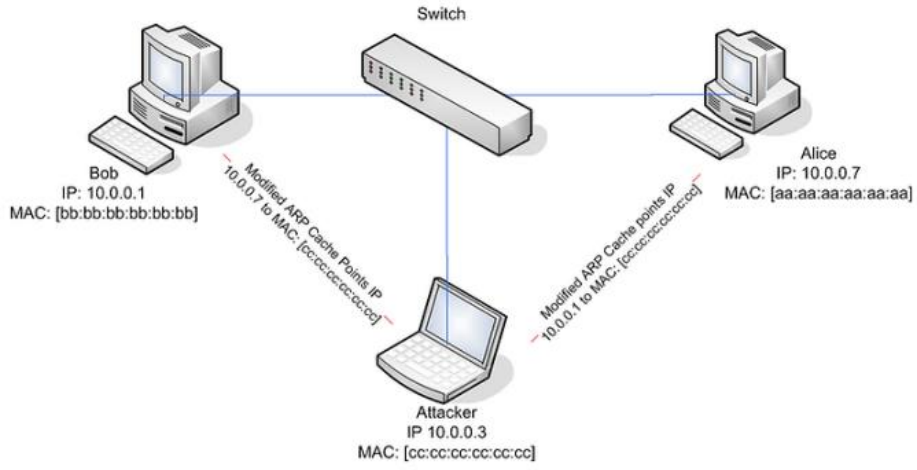
kali > netdiscover -r 192.168.100.0/24

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

22 Captured ARP Req/Rep packets, from 10 hosts.   Total size: 1320
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.42.1    00:80:ae:b6:ef:7f     11     660  HUGHES NETWORK SYSTEMS
192.168.42.2    94:6a:b0:15:41:6a      1      60  Arcadyan Corporation
192.168.42.4    70:1a:04:f4:b9:d0      2     120  Liteon Technology Corporation
192.168.42.8    30:e3:7a:55:3c:05      1      60  Intel Corporate
192.168.42.3    38:f7:3d:31:71:52      1      60  Amazon Technologies Inc.
192.168.42.15   00:0c:29:8f:ca:00      1      60  VMware, Inc.
192.168.42.11   88:b6:ee:7c:eb:ab      2     120  Dish Technologies Corp
192.168.42.22   88:b6:ee:7c:eb:ab      1      60  Dish Technologies Corp
192.168.42.6    00:7c:2d:b4:0e:3b      1      60  Samsung Electronics Co.,Ltd
192.168.42.10   88:b6:ee:7c:eb:ab      1      60  Dish Technologies Corp
```

As you can see above, netdiscover enumerates every system on the network with its IP address, MAC address, and vendor of the network interface (NIC).

---> ---> ---> ARP VULNERABILITIES

---> ARP can be used for **MITM attacks**. But how? As we know ARP can be used for assigning IP address. Attackers can send out ARP requests so that their computer is seen as the location that the target computer is trying to reach and by doing so attacker can place itself in the middle of the conversation. By doing this, an attacker can eavesdrop or even alter the conversation. This is known as **arpspoofing**

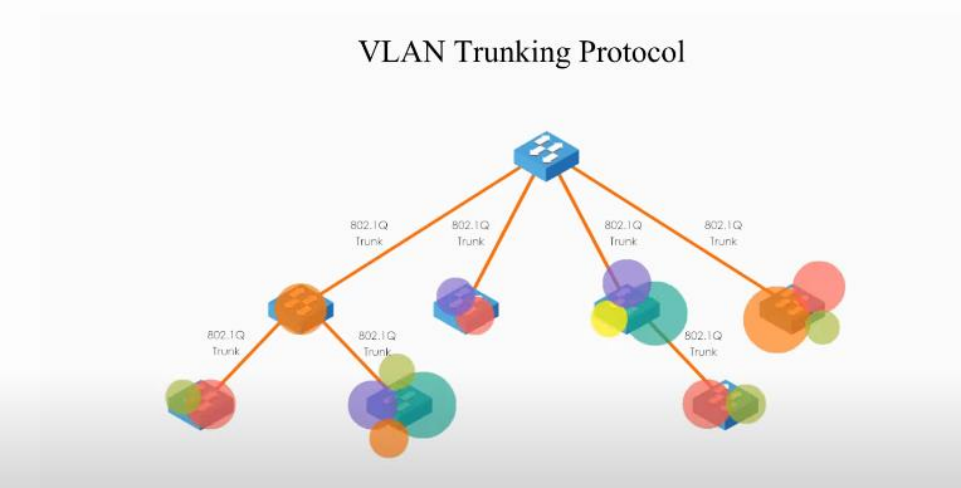


---> How do we explore the other devices on the network? By sending them ARP requests.

TRUNKING refers to the practice of combining multiple network links or ports into a single logical channel to increase bandwidth.

VTP: VLAN Trunking Protocol

---> VTP aims to manage all configured VLANs consistency across a switched network.



---> Imagine a scenario where you have dozens of switches, and those switches have different VLANs. If you try to configure all of them yourself, you'll have an issue on your hand.

---> With VTP, we can create VLANs on one switch: VTP server and all of the other switches will synchronize themselves.

---> We can add, delete or edit a VLAN in VTP server and VTP Server then distribute all these configurations to other VTP clients.

---> There are 3 modes of VTP you can choose:

->First mode is **SERVER** mode: It can create VLANs, it can send updates & advertises VTP databases.

->Second mode is **CLIENT** mode: Can't create VLANs, it can send updates & advertises VTP databases

->Third mode **Transparent** mode: Can create local VLANs, doesn't update or advertise.

---> **Requirements**:

-->Links must be trunks

-->Same VTP domain name

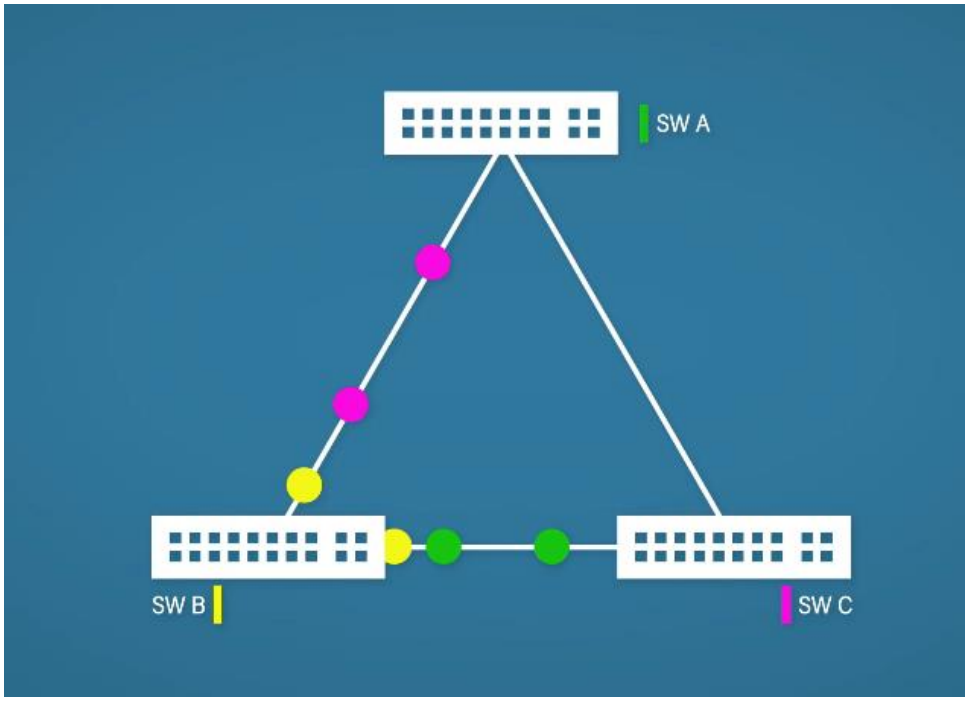
-->Same VTP password(optional)

---> Each time you make a change in VLANs the revision number increases.

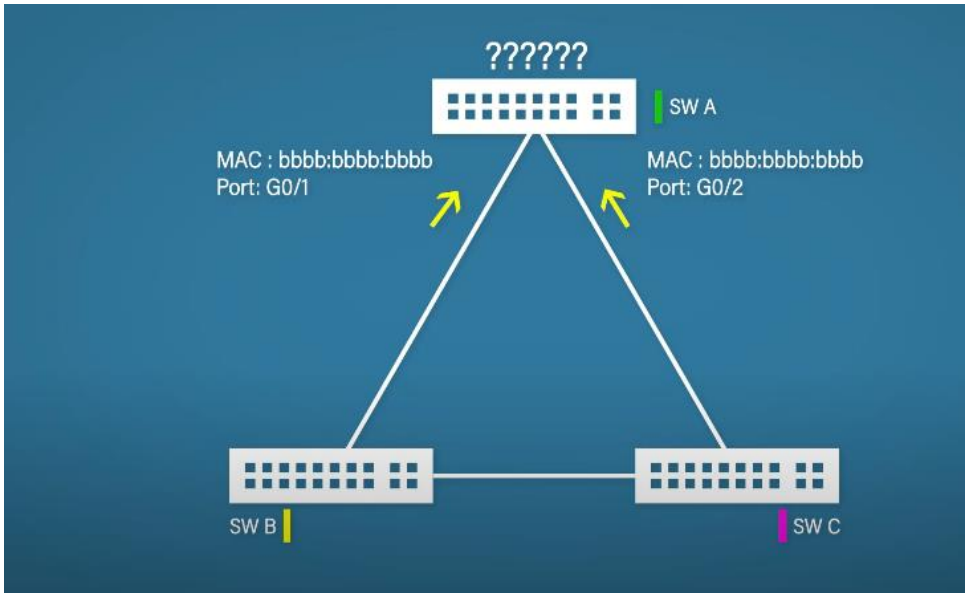
DTP: Dynamic Trunking Protocol

---> Is an automatic trunking protocol used by cisco. It is enabled on cisco switches by default. "switchport mode trunk" is manual

STP: Spanning Tree Protocol

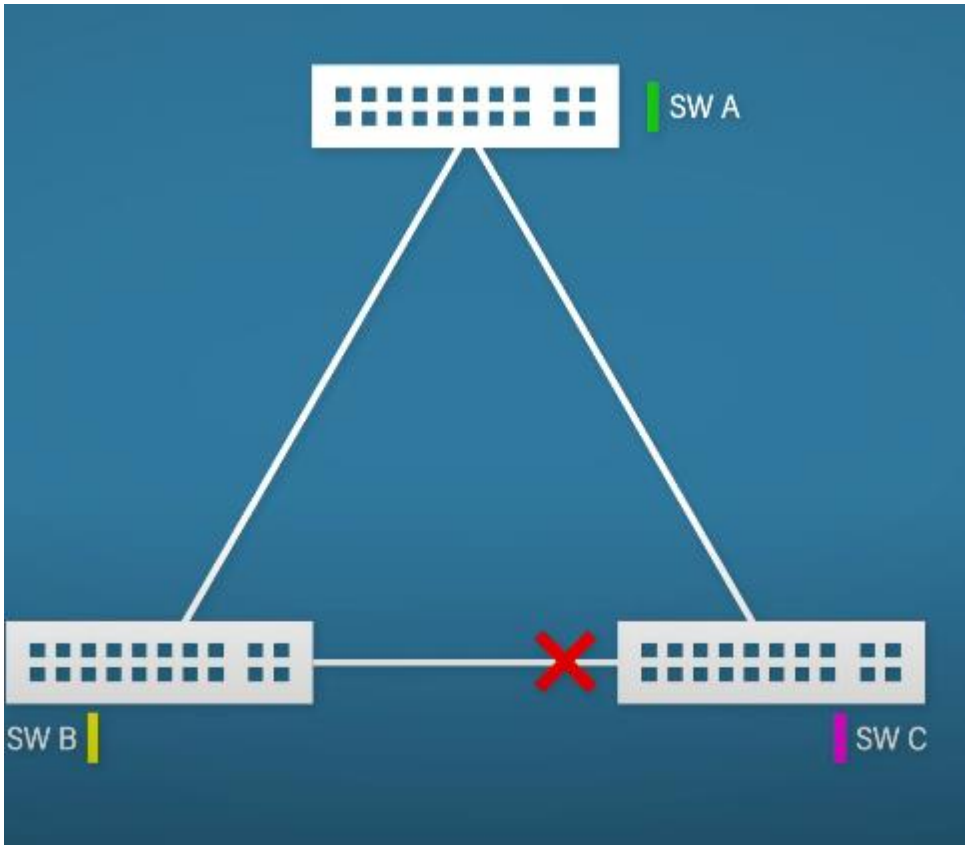


---> Assume that switch B is broadcasting a message. When it goes to switch A and C, switch A broadcasts it to switch A and switch B, and switch A broadcasts it to the switch B and switch C. And as it can be seen this causes unnecessary traffic and the traffic gets bigger and bigger what happens is known as “broadcast storm”

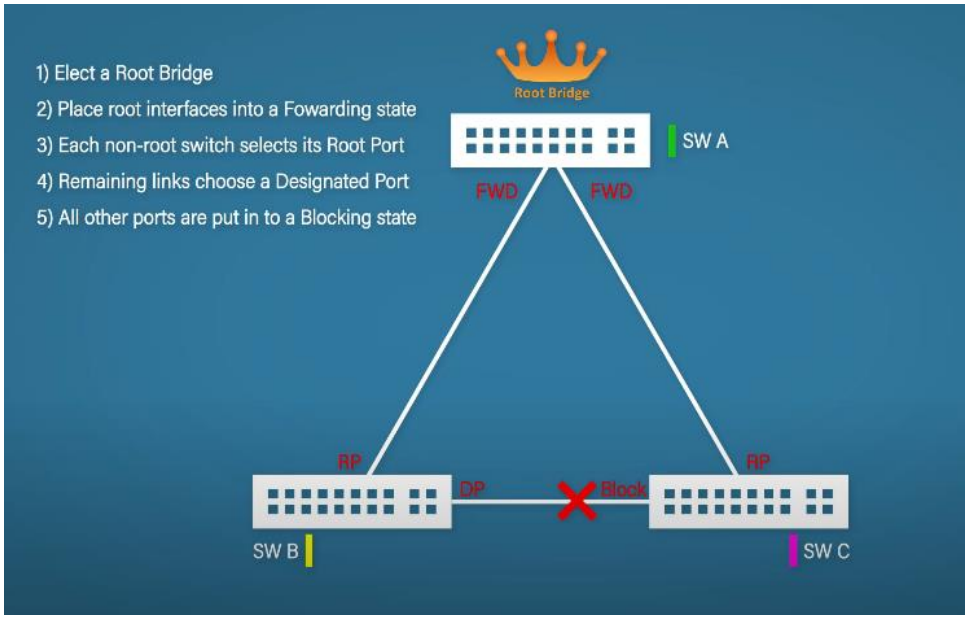


---> Switch B sends message to switch A and C and after that Switch C also sends this message to A. But A is confused. Because it had that message and MAC address but from the different Port. So, it changes its port.

---> The solution is simple



The switch C dismisses the message and only receives from the switch A. And if a problem happens on different trunks the closed trunk will be opened.



HSRP: Hot Standby Router Protocol

---> It is a proprietary CISCO protocol that provides redundancy for local subnet that provides high availablity and fault tolerance for routing in a network.

---> It allows you to configure two or more routers as standby routers and only a single router as an active router at a time.

---> It makes sure that if primary router fails, a backup router can quickly take over to minimize downtime and service disruption.

CDP: Cisco Discovery Protocol

---> It is primarily used for discovering and collecting information about neighboring network devices.

Network Layer (Layer 3)

Intro

---> **Transfers** the **network packets** from the source all the way to the destination.

---> Responsible for **packet forwarding** including **routing**. And route selection.

---> Responds to **service request** from the **transport layer**.

---> Issues **service requests to the data link** layer.

Functions of Network Layer:

---> **Connectionless communication**.

---> **Host addressing**.

---> **Message forwarding**.

---> Some protocols such as OSPF and RIP determines the “optimal” path data should take to reach a device.

Internet Protocol

---> ---> IP is responsible for:

---> **Addressing hosts**.

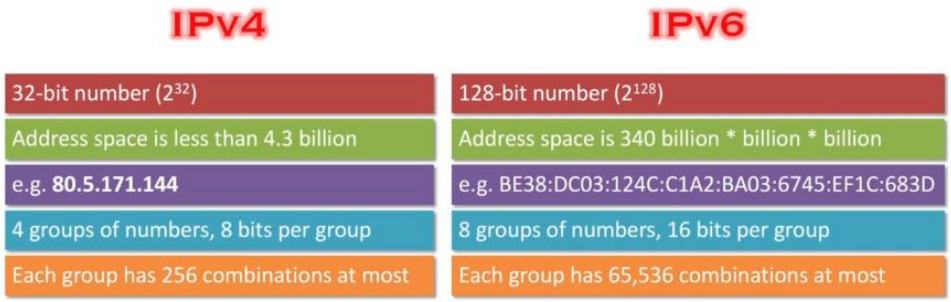
---> Encapsulating data into packets

---> Routing packets from a source to a destination.

---> **IP is connectionless**. It doesn't care if the packet has reached to the destination.

ipv4 is 32 bit and ipv6 is 128 bits long.

---> The difference between ipv4 and ipv6 is:



IPv4 Addressing System

---> Even though IPv6 is introduced this protocol still produces most of the communication.

---> Each address consists of 4 octets. 32 bits in total.

Example:

131.	107.	1.	12.
10000011	01101011	00000001	00001100

---> Each octet can be any number from 0 to 255.

---> Computers in the same network shares 3 of the 4 numbers.

IPV4 Subnetting

Type	Purpose	Explanation	Example
Network Address	This address identifies the start of the actual network and is used to identify a network's existence.	For example, a device with the IP address of 192.168.1.100 will be on the network identified by 192.168.1.0	192.168.1.0
Host Address	An IP address here is used to identify a device on the subnet	For example, a device will have the network address of 192.168.1.1	192.168.1.100
Default Gateway	The default gateway address is a special address assigned to a device on the network that is capable of sending information to another network	Any data that needs to go to a device that isn't on the same network (i.e. isn't on 192.168.1.0) will be sent to this device. These devices can use any host address but usually use either the first or last host address in a network (.1 or .254)	192.168.1.254

Subnets use IP addresses in three different ways:

1.) Identify the network address

2.) Identify the host address

3.) Identify the default gateway

---> ---> ---> ---> Classful Networks

---> <u>Classes A B and C which provide unicast networking.</u>

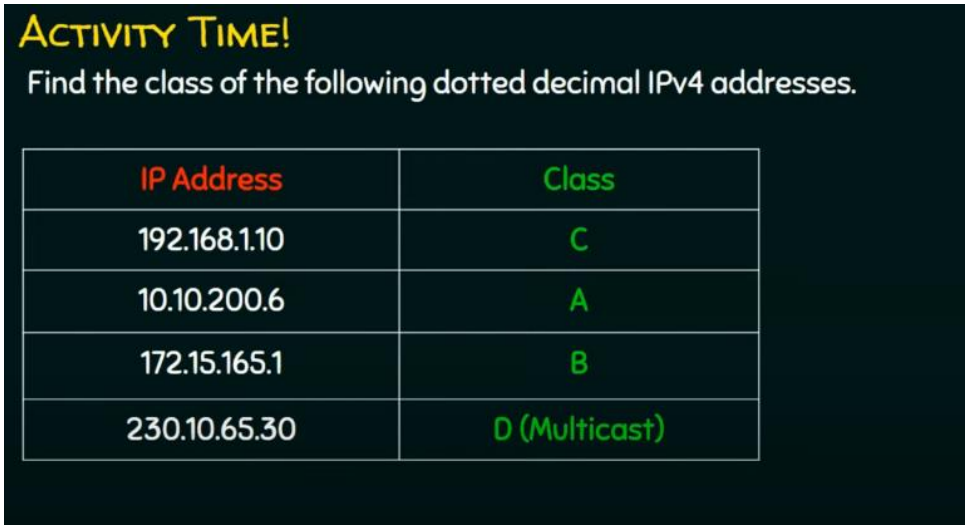
Given an IP address its class can be determined with most significant bits of the first octet.

---> <u>Class D is for multicast networking</u>.

---> <u>Class E is reserved for future uses</u>.

---> In a class A address, the first octet is a network portion, so class A has a major network address. The next 24 bits are for network manager to divide into subnets and hosts as (s)he sees fit.

---> ---> Neso Academy



PART 2 IN NESO ACADEMY

---> To know who all in the network are then the IP address must be accompanied with another parameter which is subnet mask. Subnet mask tells who all in the network are and IP address tells the identity of the device in the network.

---> ---> ---> ---> Subnet Mask

---> Subnet Mask identifies the network portion and the host portion

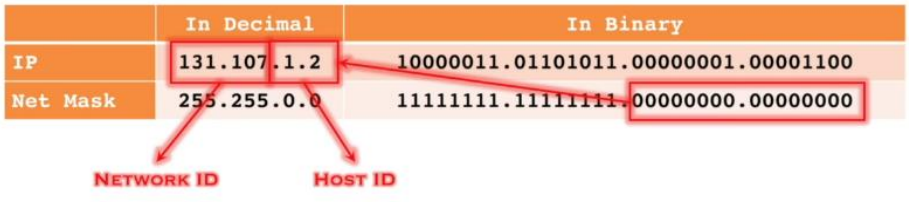
---> Default Masks (Natural Masks)

For Class A = 255.0.0.0

For Class B = 255.255.0.0

For Class C = 255.255.255.0

---> We can customize the subnet mask to divide a network in many small portions.



---> All devices in this network shares the same Network id and the same network mask.

---> In Network Mask any binaries consisting of 1s represent the Network id and any bits that are 0s represents the Host id.

---> We can customize the network mask to divide the network into smaller portions.

---> ---> ---> ---> IPV4 Subnetting

---> Subnetting allows us to use multiple logical networks.

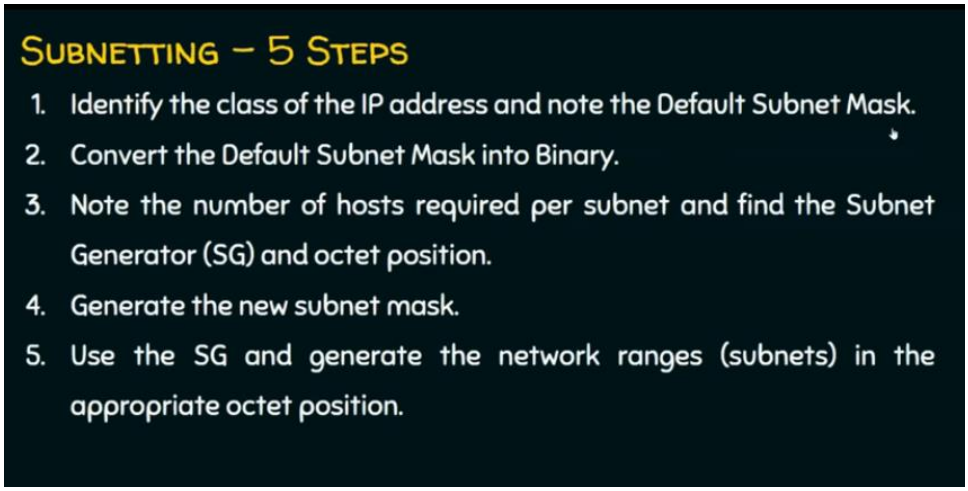
---> Subnetting is the process of designating some higher order bits from the host part as part of the network prefix and adjusting the subnet mask appropriately.

---> We can create subnets by moving two bits from host part of the network prefix.

	Binary Form	Decimal Form
IP Address	11000000.10101000.00000010.10000010	192.168.2.130
Subnet Mask	11111111.11111111.11111111.11000000	255.255.255.192

---> ---> ---> ---> Neso Academy Subnetting Solved Problem

---> Subnetting in 5 steps.



---> Subnet the ip address 216.21.5.0 into 30 hosts in each subnet.

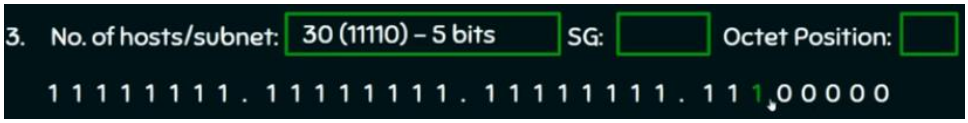
Step 1: This is a class C network and it's default subnet mask is = 255.255.255.0

Step 2: 255.255.255.0 = 11111111. 11111111. 11111111. 00000000

Step 3: Number of hosts/subnets is given to us as 30 (11110). Please note that to get 30 we need 5 bits. So, we need at least 5 bits. We are going to reserve 5 bits in subnet mask (We will reserve Least significant bits). And we are filling the Most significant 3 bits with 1s.

So, our subnet mask is 11111111. 11111111. 11111111. 1110000

Subnet Generator (SG) is the first 1 we are encountering from the right. Our SG is 32. And isince it is in the octet 4 its octet position is 4.



Step 4: Genereta the new subnet mask

New subnet mask is 11111111. 11111111. 11111111. 11100000 which is 255.255.255.224 or/27

Step 5: Use the SG and generate the network ranges(subnets) in the appropriate octet position.

216.21.5.0 This is the original one. And the octet position is 4 so we are only going to make changes in the 4th octet.

So the IP addresses are

216.21.5.0 -	216.21.5.31	First subnet will start from here.
216.21.5.32 -	216.21.5.63	Second octet will start from here
216.21.5.64 -	216.21.5.97	Third octet will start from here
216.21.5.96 -	216.21.5.127	Fourth octet will start from here
216.21.5.128 -	216.21.5.159	Fifth octet will start from here

and so on...

---> ---> ---> ---> IPV4 Shortage

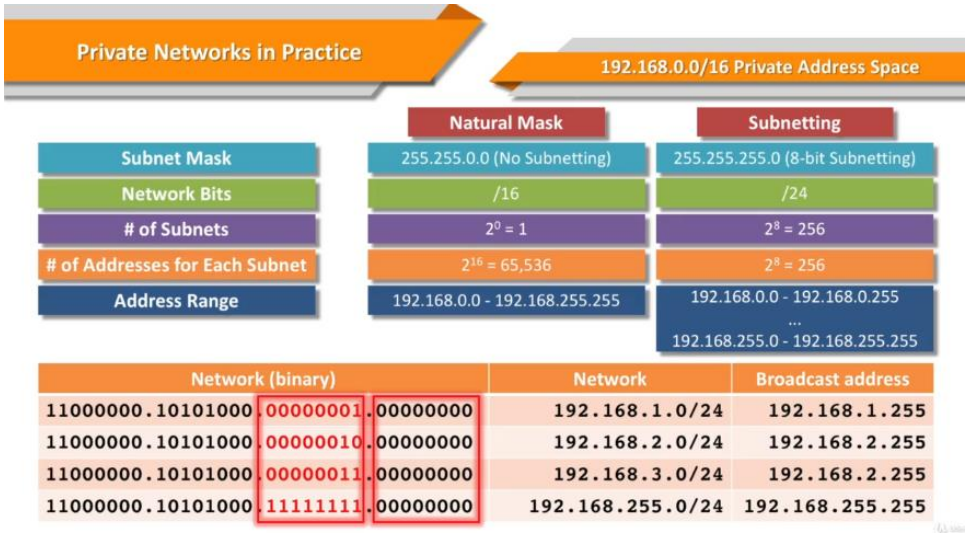
The maximum amount of ipv4 is 2^32 in theory which is 4 billion. But there is much more internet connected devices than that.

So, the long-term solution to that is to turn into IPv6

Short Term solution is Private Networks & Network Address Transmissions (NAT)

Private Networks

---> Generally an IP address that starts with 169 means our computer wasn't able to obtain an IP address from our DHCP server. Meaning our router is wedged and needs to be rebooted or our Wi-Fi isn't working.



---> Private addresses include:

192.168.0.0 - 192.168.255.255

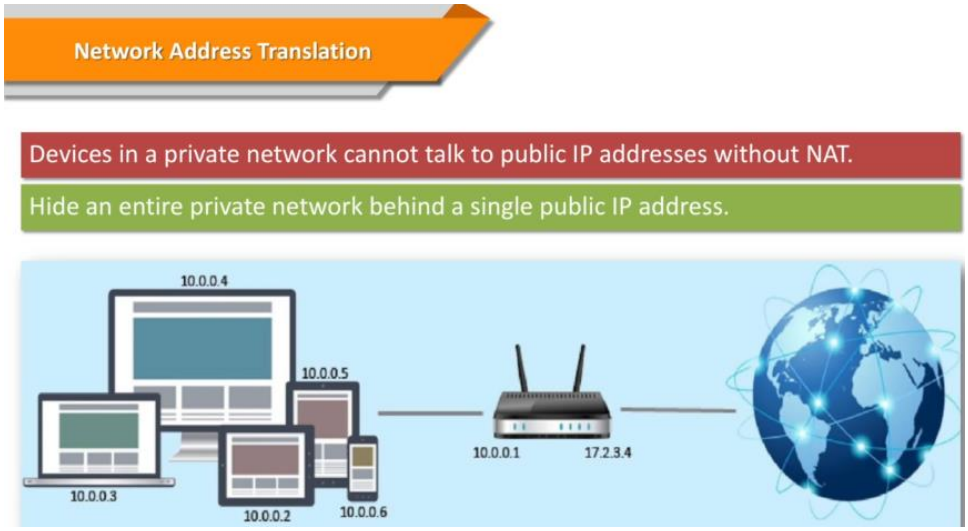
10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.16.255.255

NAT

---> The basic principle of NAT is that many computers can “hide” behind a single IP address. (They kind of have to because there are not enough IP addresses out there.) NAT acts like a gateway between internal and external networks

---> **NAT translates internal IP address into external IP address**. (IP in the network to IP on the internet)



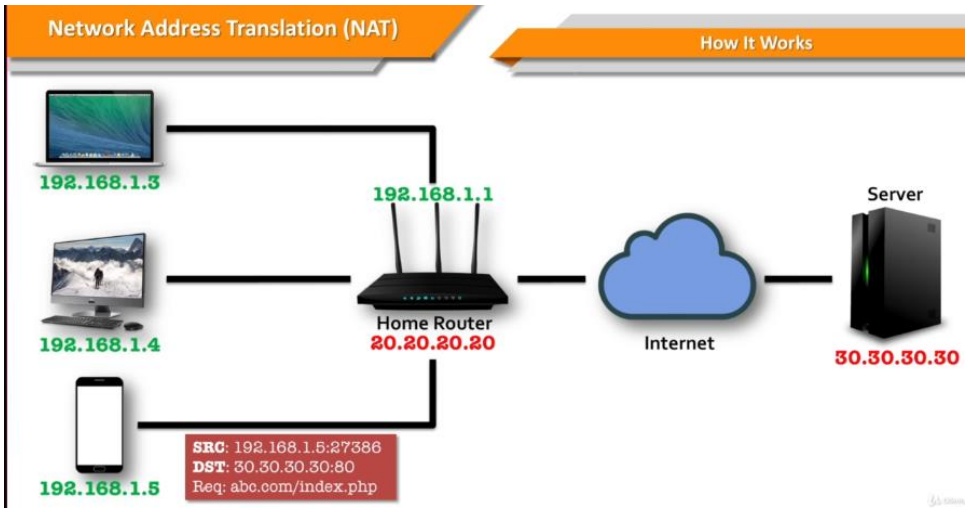
---> Workstation inside a private network makes a request to a computer on the internet. The switches, routers realizes that the request is not for a resource from the network. So, they send request to router, router seizes request with internal ip. And makes the request by its public address

to the internet and returns the response from the internet resource to the computer inside the private computer.

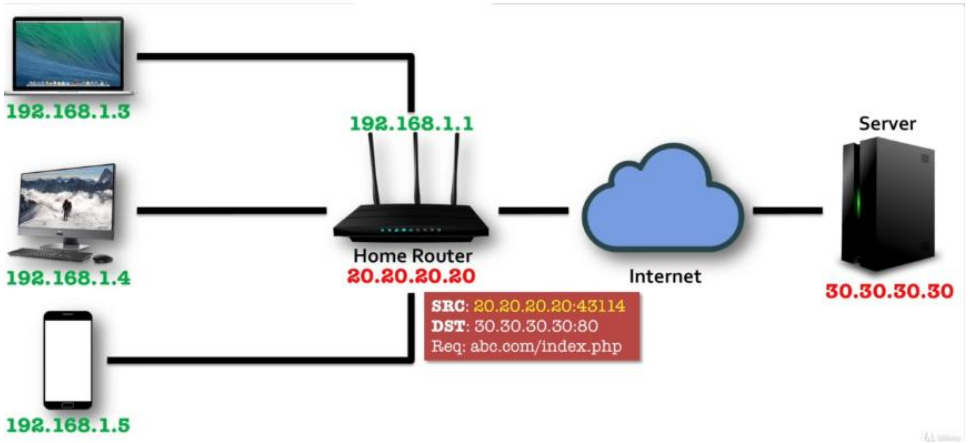
---> It is sending information to the router.

---> The 10.0.0.3 and stuff are private ip addresses.

Assume that 20.20.20.20 assigned to the home router. Our devices inside the home network also have private IPs, they are assigned by router and not accessible from the internet. Public IP addresses are red, and privates are green in the photo below.

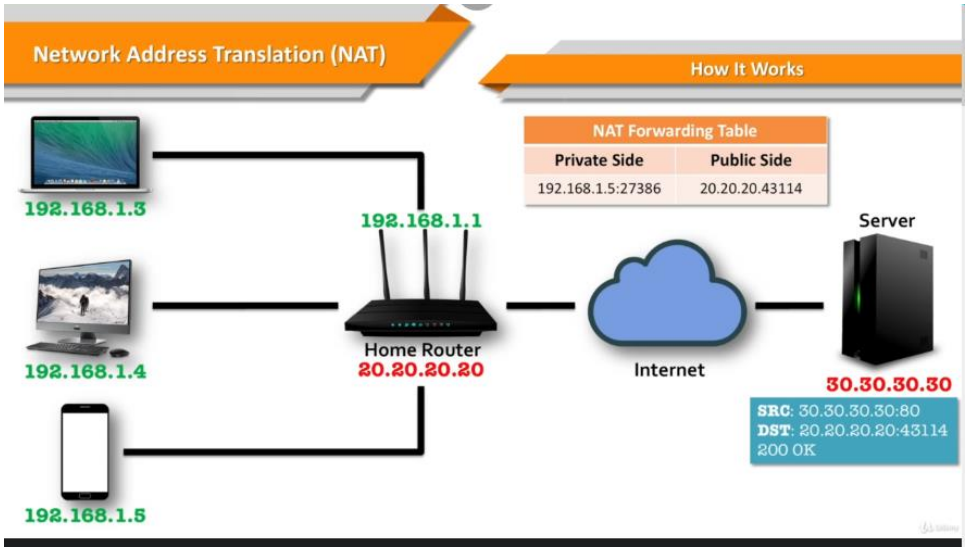


---> Let's say phone wants to go to abc.com. To reach the page, the smart phone has to go through home router. it cannot go to the internet with its private IP address because it is private. So, it is unreachable for internet.



---> So when it comes to the home router, the home router changes its source IP address and creates a record in NAT forwarding table.

---> Then packet travels through the internet and reaches the server. And server creates a packet back to the home router using that home routers public IP address.



---> When the packet gets to the home router, our home router looks at the NAT forwarding table and changes the destination address and then the smartphone receives the packet.

IPv6

---> Developed to deal with the problem of IPv4 address exhaustion.

---> IPv6 has some features IPv4 doesn't have.

---> SUCH AS:

->Simplified header,

->End-to-end connectivity this means every host can reach other hosts without any need of NAT and stuff.

->Auto configuration,

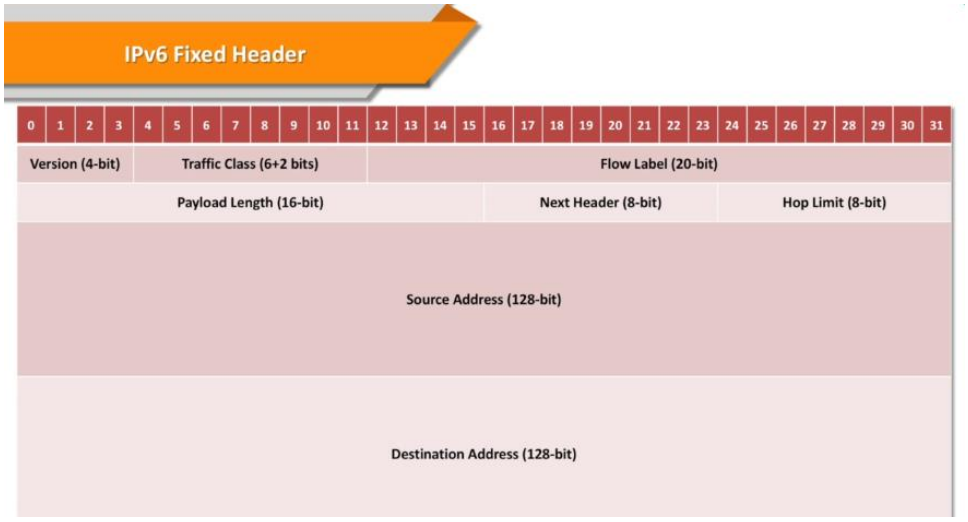
->Faster routing/forwarding,

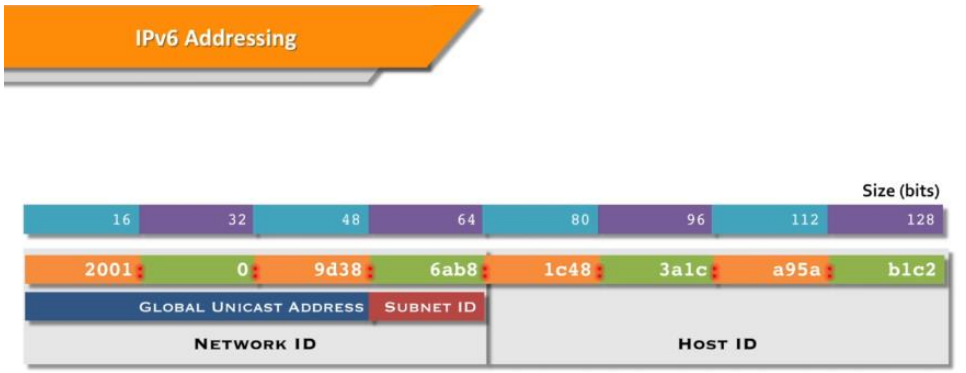
->IPSec = Optional but makes it more secure.

->No Broadcast: ipv6 doesn't have broadcast support but has multicast.

->Mobility

->Extensibility.



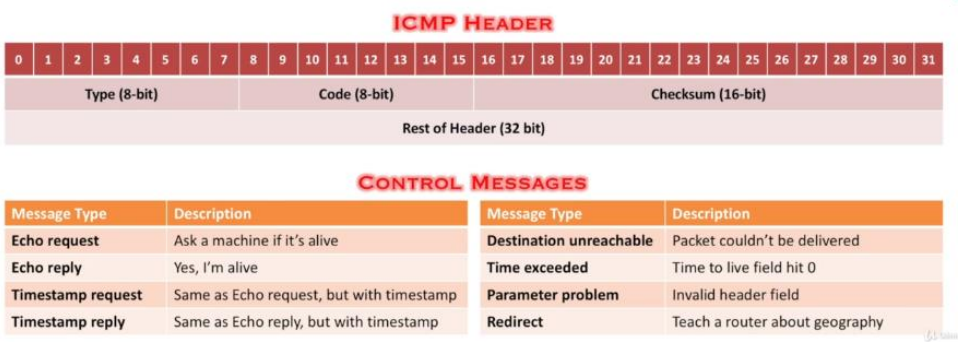


ICMP (Internet Control Message Protocol)

---> **Error checking & reporting** protocol that **works with IP layer**.

---> Purpose is to provide feedback about problems, **not making IP reliable**.

---> Its most common function is **ping** utility.



Control Messages

Message Type	Description	Message Type	Description
Echo request	Ask a machine if it's alive	**Destination unreachable**	Packet couldn't be delivered
Echo reply	Yes, I'm alive	**Time exceeded**	Time to live field hit 0
Timestamp request	Same as Echo request, but with timestamp	**Parameter problem**	Invalid header field
Timestamp reply	Same as Echo reply, but with timestamp	**Redirect**	Teach a router about geography

Ping

Ping is used to test the connection

Traceroute

Another command that uses ICMP is traceroute. The internet is made up of a lot of different servers. Traceroute not only allows you to check connections but it also allows you to check intermediate ste

```
                ~$ traceroute google.com
traceroute to google.com (216.58.205.46), 30 hops max, 60 byte packets
 1  _gateway (172.16.255.254)  14.883 ms  15.401 ms  15.551 ms
 2  193.60.160.253 (193.60.160.253)  1.464 ms  1.872 ms  2.026 ms
 3  193.60.168.92 (193.60.168.92)  3.084 ms  4.093 ms  4.814 ms
 4  ge-0-3-2.dund-ban1.ja.net (146.97.128.85)  4.768 ms  4.253 ms  4.715 ms
 5  ae1.dund-ban3.ja.net (146.97.64.97)  10.320 ms  5.114 ms  10.589 ms
 6  ae24.leedaq-sbr2.ja.net (146.97.37.181)  11.160 ms  10.855 ms  10.766 ms
 7  ae29.lowdss-sbr1.ja.net (146.97.33.50)  11.992 ms  11.048 ms  10.746 ms
 8  ae31.londtw-sbr2.ja.net (146.97.33.30)  13.558 ms  13.245 ms  13.561 ms
 9  ae28.londtt-sbr1.ja.net (146.97.33.61)  13.541 ms  13.229 ms  11.410 ms
10  72.14.205.74 (72.14.205.74)  15.143 ms  14.607 ms  13.865 ms
11  74.125.242.97 (74.125.242.97)  13.263 ms 74.125.242.65 (74.125.242.65)  12.553 ms  12.904 ms
12  172.253.71.191 (172.253.71.191)  13.943 ms  12.833 ms 172.253.71.189 (172.253.71.189)  12.631 ms
13  lhr48s23-in-f14.1e100.net (216.58.205.46)  13.227 ms  12.258 ms  12.482 ms
```

ps.

As you can see here it takes this computer 13 hops to get to the google.

Windows equivalent of traceroute is tracert.

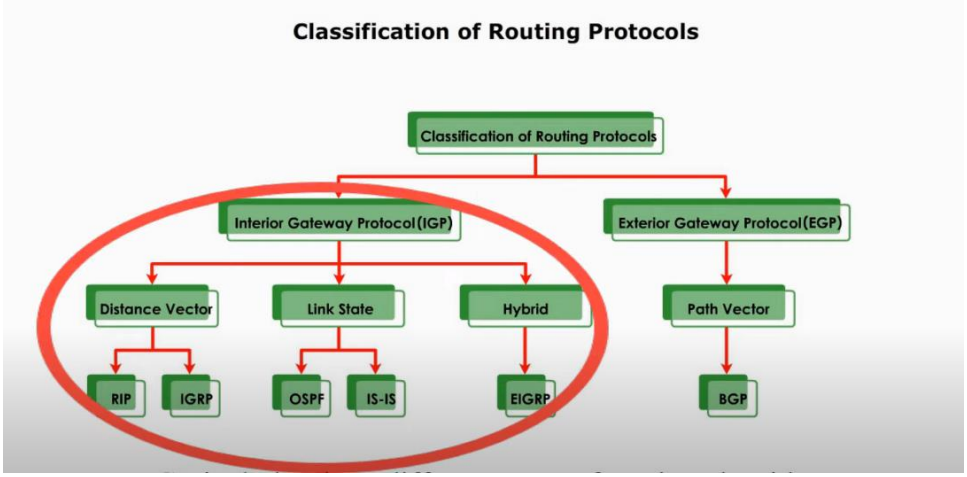
whois

```
                ~$ whois bbc.co.uk

   Domain name:
       bbc.co.uk

   Data validation:
       Nominet was able to match the registrant's name and address against a 3rd party data source on 12-Jun-2014

   Registrar:
       British Broadcasting Corporation [Tag = BBC]
       URL: http://www.bbc.co.uk

   Relevant dates:
       Registered on: before Aug-1996
       Expiry date:  13-Dec-2025
       Last updated:  29-Oct-2016

   Registration status:
       Registered until expiry date.

   Name servers:
       ns3.bbc.co.uk             156.154.66.17  2610:a1:1015::17
       ns3.bbc.net.uk
       ns4.bbc.co.uk             156.154.67.17  2001:502:4612::17
       ns4.bbc.net.uk

   WHOIS lookup made at 02:22:04 07-Mar-2020
```

IPsec

---> IP Security protocol, designed to provide secure communications between systems. It can encrypt and authenticate network transmission.

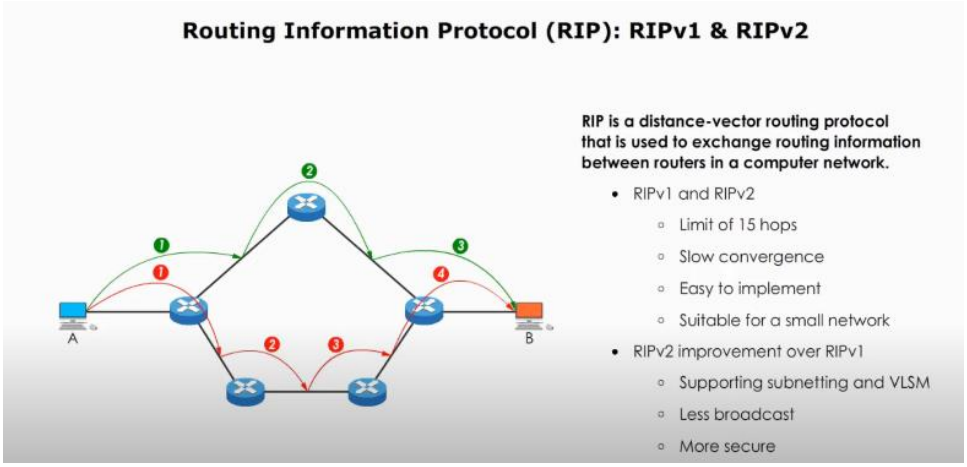
Routing Protocols



---> Distance vector routing protocols are a class of routing protocols that determine the best path.

RIP (Routing Information Protocol)

---> RIP is a distance-vector routing protocol that is used to exchange routing information between routers in a computer network. Works for internal networks



OSPF (Open Shortest Path First)

--->

Transport Layer (Layer 4)

Introduction

---> Transport layer is the layer where system **reliability and quality are ensured**. It transports data between network devices.

---> Basic functions are:

-> Split up data into smaller units. (**Segmentation**). Where the target computer later re-constructs.

-> Pass the units to the network layer and accept data from layer above. (**Service Addressing**)

-> Make sure that all the units arrive correctly at the other end. (**Error checking**)

-->UDP and TCP are here

---> ---> ---> ---> FUNCTIONS

---> Service Point Addressing = Which is the port address. Which brings each packet to the correct process. Like HTTP or HTTPS

---> Segmentation and Reassembling

---> Connection Control: Connectionless (UDP) and connection oriented (TCP).

---> Flow Control: Checks if the transfer of data is sufficient for management. For fast sender or other things. Prevents buffer overrun

---> Error Control: Done via re-transmission

TCP-Transmission Control Protocol

---> ---> TCP BASICS

---> Is a "Connection oriented" protocol that works in the transport layer.

---> It determines how to fragment the data.

---> Sends and accepts data from network layer.

---> Manages flow control

---> Makes sure data is error free.

---> When a web server sends an HTML file to a client it uses the HTTP protocol to do so. The http protocol asks TCP to provide connection so it can send file. The TCP stack divides the file into packets and numbers them and then forwards packets individually to the IP layer. TCP of client computer waits until all of the packets and sends a message about the ones it received and asks for the ones who it did not received.

---> ---> ---> TCP 3 WAY HANDSHAKE

---> TCP Flags

Used within TCP packet transfers to indicate a particular connection state or provide additional information.

-> SYN = Synchronization is the first step of starting the communication between 2 computers

-> ACK = Acknowledges the receiver, received the flag.

-> RST = Gets send back to the sender, when a packet is sent to a host that was not expecting it

-> FIN = Finished, means no more data to send. This is sent at last.

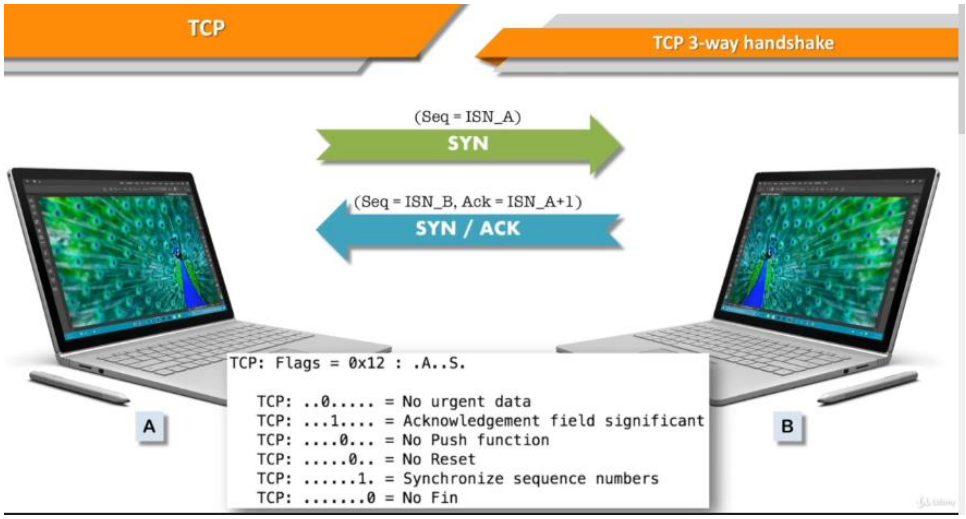
->PSH = Push. Tells the receiver to process these packets when they are received instead of buffering them.

-> URG = Notifies the receiver to process the urgent packets before processing all other packets.

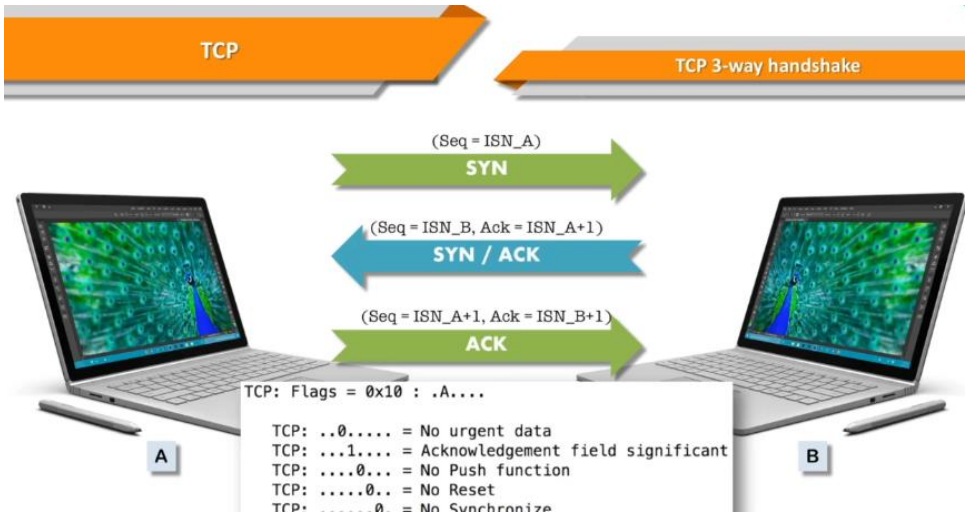
---> All TCP connections begins with, TCP 2-way handshake. Which is used to exchange sequence numbers so the lost packets can be re-transmitted.

1.) SYN is 32 bits long. ACK is also 32 bits long.

2.)

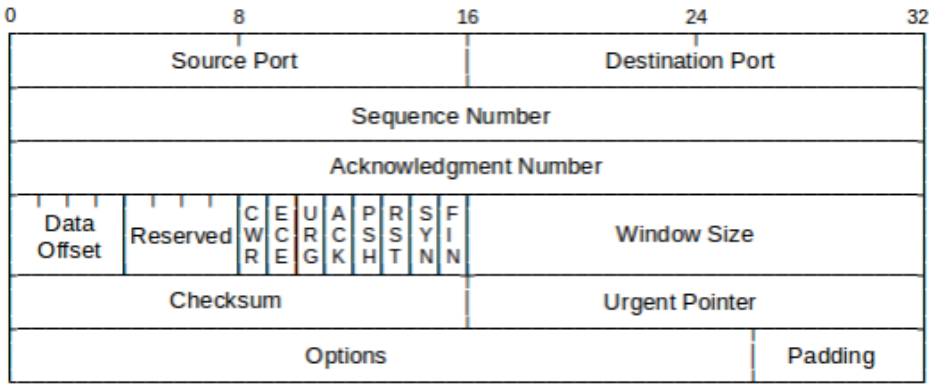


3.)



And after that the data transfer begins.

---> ---> ---> OTW



ROW 1: Source Port / Destination Port: These fields determine what port the communication came from (source) and to where it is going to (destination).

ROW 2: Sequence Number: It is generated at source machine's TCP stack, and it is used to make certain that packets are arranged in the proper sequence when they arrive. It is important for defeating MitM attacks.

ROW 3: Acknowledgement number: This basically says, "I received the packet with the sequence #". If the sender doesn't receive acknowledgment number back in a fixed amount of time, it'll resend the packet to make certain receiver gets the packet. In this way, TCP is reliable.

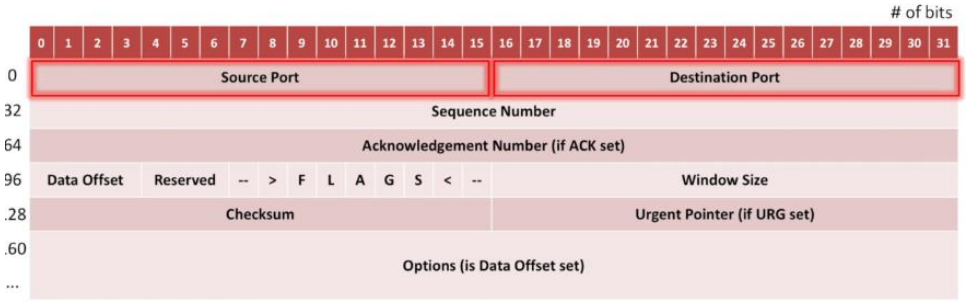
ROW 4: Window size: Its role is to communicate the size of the windows that the TCP stack has to buffer packets.

ROW 5: Checksum: Checks errors.

URG Pointer: Points to the last byte of the sequence number of urgent data.

ROW 6: Paddling: Is necessary for bringing TCP header to a multiple of 32 bits.

TCP Header



---> Sequence Number and Acknowledgement number makes sure data arrives reliably and in order. if a packet is lost TCP will send it again.

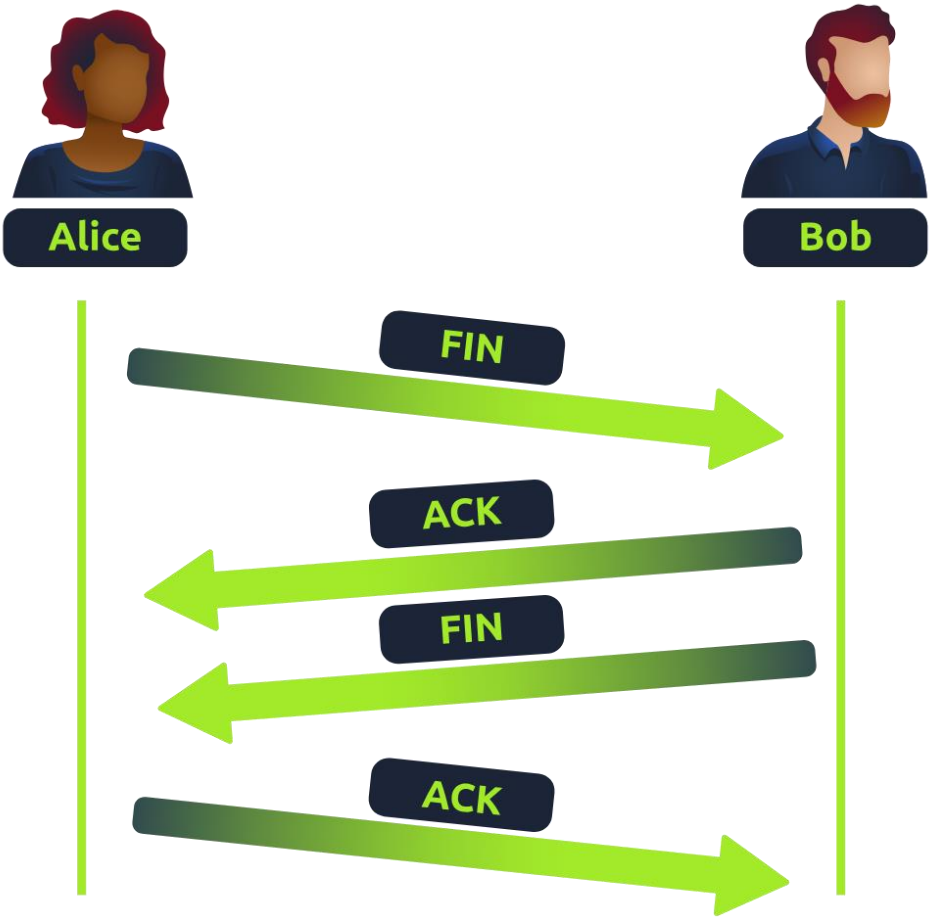
---> Data offset specifies the size of a TCP header. The size range is between 5 and 15 words.

---> Window Size = They regulate how much data they send to a receiver requiring an acknowledgement in return. So, if windows size is small this means network data transfer is slow. And if it is large, the network link can become saturated, or the receiver may not be able to process it

TCP Closing a Connection

---> TCP'll close a connection once a device has determined that the other device has successfully received all of the data. TCP reserves system resources on a device, closing TCP connections as soon as possible is a good practice.

---> To initiate a closure, the device will send a "FIN "packet to the other device. Of course, the other device will also have to acknowledge this.



UDP-User Datagram Protocol

---> ---> ---> General Properties Of UDP

---> No Handshake = UDP is a connectionless communication

---> No Error Checking = Lost Packets are not sent again.

---> Faster Data Transfer = Suitable for Multimedia

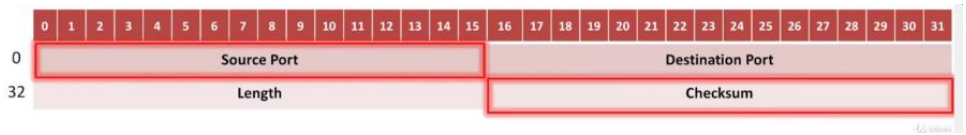
---> No Validation = IP spoof is possible

---> Some Usages are DNS, DHCP, Multimedia

---> UDP can cost in terms of additional data overhead. Packets can be lost or received out of order.

---> No guarantee of delivery and protection.

---> UDP is suitable for purposes where error checking and correction are not necessary or important such as gaming voice and video communications, which can sever some data loss without so much trouble.



---> Source port and checksum is optional.

---> ---> TCP vs UDP

---> TCP tries to preserve sequence and if packets are lost it re-transmits it. Because of these things it is slow.

---> UDP doesn't make any attempt for reliable data transfer. Because of that it is fast.

Application Layer (Layer 5-7)

Introduction

---> Application layer standardized communication. Establishes Host-To-Host data transfer channels and manage the data exchange.

SESSION LAYER (LAYER 5)

---> The Session Layer is responsible for **establishing**, **maintaining**, and **terminating** communication sessions between devices. It manages the **dialog control**, **synchronization**, and **data exchange** between applications on different devices

---> When data is transferred from presentation layer (layer 6) to session layer (layer 5) it'll begin to create a connection to the other computer.

---> When a connection is established, a **session** is created.

NetBIOS- Network Basic Input/Output System

---> Is an API

PRESENTATION LAYER (LAYER 6)

---> Layer 6 of the OSI model is the Presentation Layer. The Presentation Layer is responsible for **data translation (main purpose)**, **encryption**, **compression**, and other transformations necessary to ensure that data from the application layer of one system can be properly understood by the application layer of another system.

APPLICATION LAYER (LAYER 7)

---> It works with applications to provide interface for them (GUI's work here.). Provides an interface for transmitting data.

DNS- Domain Name System

---> DNS works like **phonebook of the internet**. It translates human-friendly names into IP address. Because of DNS instead of <ip_address_of_google> you see www.google.com

---> It is a platform-independent protocol. It can be used in UNIX, LINUX, Windows, MAC etc.

---> When you type a URL, the URL goes to the DNS server first and DNS server looks up to the lookup table to resolve the IP address and returns this info to us.

---> DNS server contains the host names and their corresponding IP addresses.

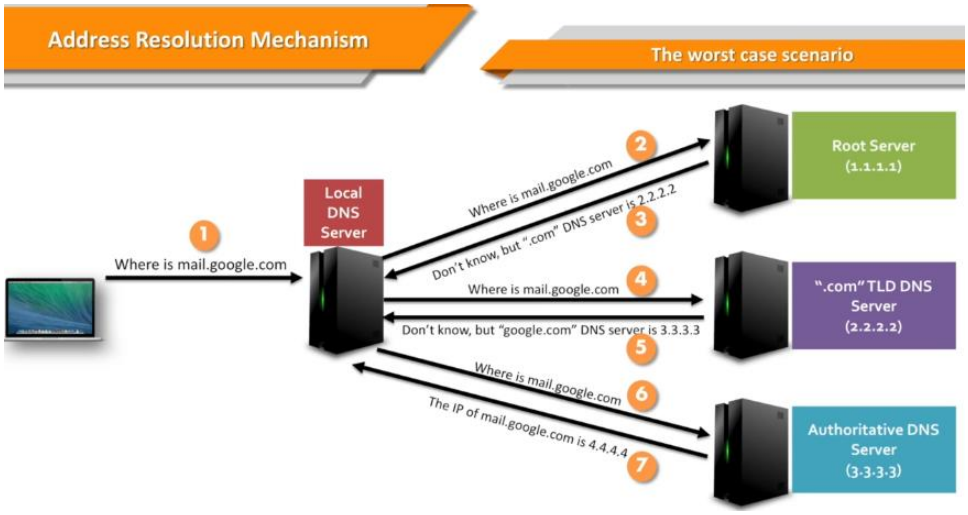
---> ---> ---> DNS Architecture

---> Client/Server network communication system

---> Forward DNS lookup is a standard lookup. When you write an URL and get IP address.

---> **Reverse DNS lookup** is the reverse of it. You are writing the IP address of a webservice and get the name of it is reverse DNS lookup.

---> The DNS servers communicate with each other, and all of the directories are distributed in the world.

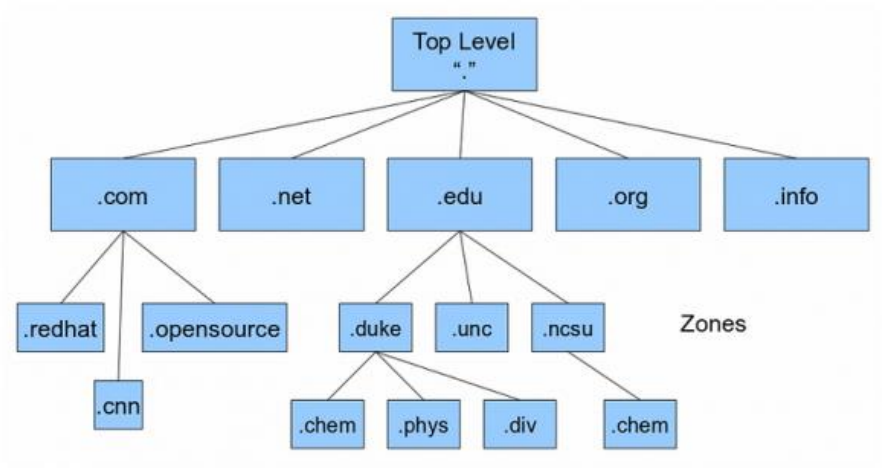


---> ---> ---> OTW DNS

---> It basically turns human-friendly URL names into IP addresses.

DOMAIN NAMES

Domain names must be registered ICANN (Internet Corporation for Assigned Names and Numbers). Top level domain names include .com, .edu, .org.



DNS works in a hierarchical manner. Top level domains can have sub-domains and that sub-domains can have sub-domains of themselves. They are called Secondary Level Domain or SDL if under .com.

Fully Qualified Domain Name

Refers to absolute domain name. A FQDN specifies its location from the absolute root of the DNS system.

Host Files

When internet was new all of the domain names could have fit in a small text file. This single text file is referred as a host file. However, since the internet is a lot larger and changing constantly, we don't have all of the hosts files with us however, we still have some. For example, in kali Linux if you write mousepad /etc/hosts you can find your own host file.

kali> mousepad /etc/hosts

```
127.0.0.1       localhost
127.0.1.1       kali

192.168.56.101   bankofamerica.com

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

On the fourth line of my hosts file here, you will see an association of the private IP address 192.168.1.114 to the domain bankofamerica.com. With this hosts file in place, whenever I enter www.bankofamerica.com in my browser, I would be directed to the IP address 192.168.56.101, rather than the actual IP address of Bank of America at 171.159.228.150

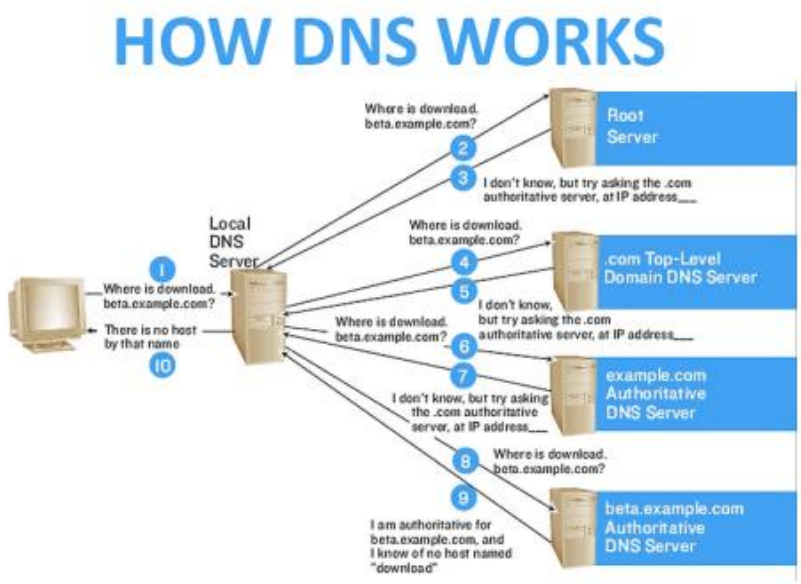
I can test by pinging bankofamerica.com.

```
kali@kali:~$ ping bankofamerica.com
PING bankofamerica.com (192.168.56.101) 56(84) bytes of data.
64 bytes from bankofamerica.com (192.168.56.101): icmp_seq=1 ttl=64 time=0.
018 ms
64 bytes from bankofamerica.com (192.168.56.101): icmp_seq=2 ttl=64 time=0.
052 ms
64 bytes from bankofamerica.com (192.168.56.101): icmp_seq=3 ttl=64 time=0.
027 ms
64 bytes from bankofamerica.com (192.168.56.101): icmp_seq=4 ttl=64 time=0.
```

When we ping bankofamerica.com, my ping is directed to the address associated with bankofamerica.com in my host file. The hosts file takes precedence over DNS queries. This can be a key bit of information when attempting to do DNS spoofing on a LAN.

How DNS Works?

DNS is both distributed and dynamic, unlike our host file. DNS doesn't rely upon one file or server, but instead upon many files across many servers around the globe. These servers are organized in a hierarchical manner. Due to this distributed nature, the DNS system is resistant to outages of one or many of these servers.



1.) As it can be seen, the user asks (queries) the local DNS server (Local Cache) to access target (download.beta.example.com) and the local server doesn't know.

2.) It then queries to the root server. The root server also doesn't know.

But it refers the local DNS server to the IP address of the authoritative server for the top-level domain (TLD), in this case **.com**.

The local DNS server will then query the TLD server for .com and it'll respond authoritative server for the domain, in this case **example.com**.

The local DNS server will then query the authoritative server for **beta.example.com**

If it has the record, it'll return the IP address and if not, it'll return "I don't know."

TO SUM UP

1.) Computer checks its local cache to see if it's already having the ip address

2.) If not found computer sends requests to the recursive DNS server. This server caches the results for popular domains.

3.) If not found recursive DNS server will send this data to the root/ (Top Level Domain) server.

---> Top level servers are split into extensions. For example, www.tryhackme.**com** would be handled by TLD server that handles .com domains. www.bbc.**co.uk** kind of domains are also handled by TLD

DNS Components

The DNS service has four (4) components;

1. DNS Cache

2. Resolvers,

3. Name servers,

4. Name space

DNS Cache

It can be 2 things. Firstly, DNS cache can be the list of names and IP addresses that you have already queried and have been resolved and are cached for you that no network traffic is generated to resolve them (For speed). Secondly, DNS server that simply performs recursive queries and caching without actually being an authoritative server itself.

Resolver

Resolvers are any hosts on the Internet that need to look up domain information.

Name Servers

These are servers that contain the database of names and IP addresses and servers' DNS requests for clients.

Name Space

Name space is the database of IP addresses and their associated names.

---> One of the problems with DNS is that, despite all its automatic resolution capabilities, entries and changes to those entries **must still be manually performed**. A strategy to solve this problem is to use **Dynamic DNS (DDNS)**, a newer system that enables hosts to be dynamically registered with the DNS server. By making changes in real-time to hostnames, addresses, and related information, there is less likelihood of not finding a server or site that has been recently added or changed.

DNS Record Types

A Record: These records resolve to *IPv4* addresses.

AAAA Record: These records resolve to *IPv6* addresses.

CNAME Record: These records resolve to *another domain name*. For example, THM's online shop has the subdomain name store.tryhackme.com which returns CNAME record shops.shopify.com

MX Record: Resolve to the address of the servers that handle the *email* for the domain you are querying.

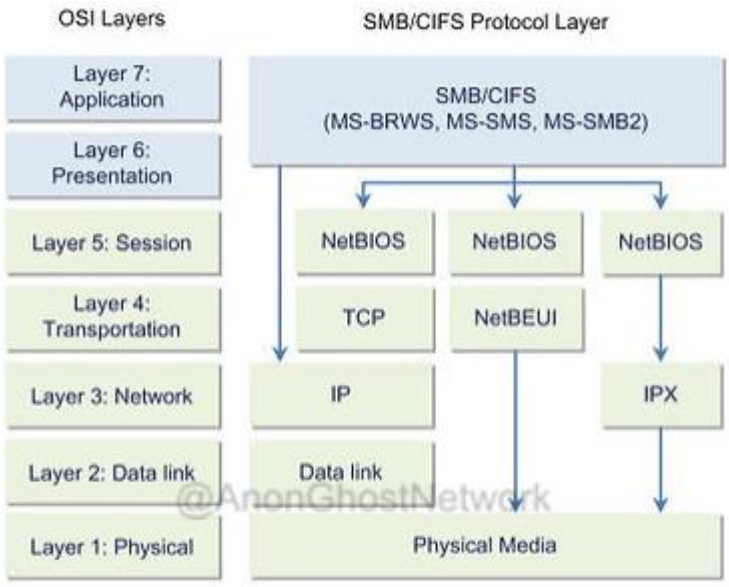
TXT Records: TXT Records are free text fields where any *text-based* data can be stored.

ÖNEMLİ NOT: Hepsini bitirmedim. Sayfa 125'e kadar geldim.

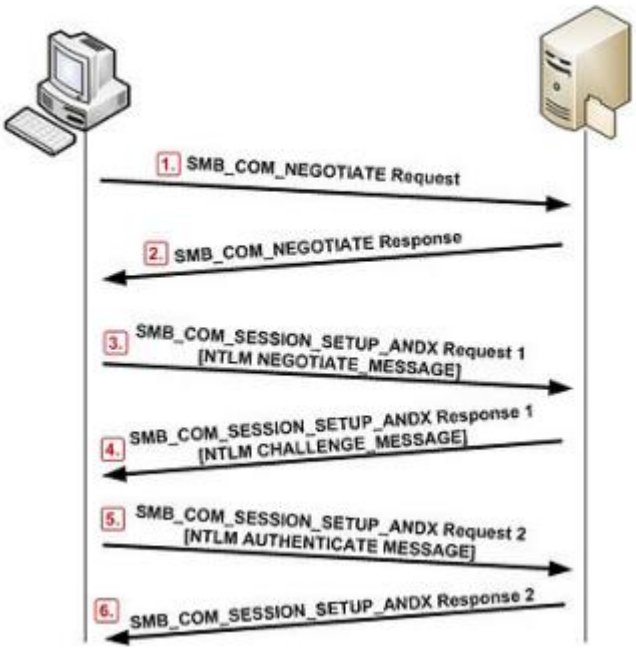
SMB: Server Message Block

---> SMB is an **application layer** (Layer 7) protocol that is used for file, port, named pipe, and printer **sharing**. It's a client-server communication protocol. It enables users and applications to **share resources across their LAN**. Meaning if a system has a file that other system needs, SMB enables the user to share their files with other users. SMB over TCP/IP uses port 139 and 445. Samba implements SMB for UNIX system types. SMB is known as **response-request protocol**.

---> Some of the tools for enumerating SMB are nmap, enum4linux and smbclient.



IPX/SPX = Internet Packet Exchange/Sequenced Packet Exchange
NetBIOS = Network
NetBEUI = NetBIOS Extended User Interface
MS-BRWS = CIFS Browser Protocol = Microsoft Browser



CIFS (Common Internet File System): CIFS is a dialect or a form of SMB. It is obsolete.

SMB Vulnerabilities: SMB in Windows and Samba in Linux has been a source for many vulnerabilities. So, if you see port 139 or port 445 open you should enumerate it throughly.

Enumerating SMB

---> Before doing anything just ping it. TTL of 64 indicates most likely a linux box. And ttl of 128 is generally common for Windows

```
root@ip-10-10-143-102:~# ping -c 3 10.10.185.8
PING 10.10.185.8 (10.10.185.8) 56(84) bytes of data.
64 bytes from 10.10.185.8: icmp_seq=1 ttl=64 time=0.711 ms
64 bytes from 10.10.185.8: icmp_seq=2 ttl=64 time=0.311 ms
64 bytes from 10.10.185.8: icmp_seq=3 ttl=64 time=0.294 ms

--- 10.10.185.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.294/0.438/0.711/0.194 ms
```

---> And then run nmap scan

```
root@ip-10-10-143-102:~# nmap 10.10.185.8

Starting Nmap 7.60 ( https://nmap.org ) at 2023-04-24 17:17 BST
Nmap scan report for ip-10-10-185-8.eu-west-1.compute.internal (10.10.185.8)
Host is up (0.00088s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 02:4B:27:CA:BC:59 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.71 seconds
```

---> To further up our research use a nmap scan with –A (all) scan that performs service version and OS detection as well as traceroute

```
nmap -A -p22,139,445 <IP>
```

You can see the details provided by the nmap scripts below:

```
Host script results:
|_nbstat: NetBIOS name: POLOSMB, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: polosmb
|   NetBIOS computer name: POLOSMB\x00
|   Domain name: \x00
|   FQDN: polosmb
|_  System time: 2023-04-24T16:20:55+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2023-04-24 17:20:55
|_  start_date: 1600-12-31 23:58:45
```

---> enum4linux will default to the –a scan.

```
enum4linux <IP>
```

```
root@ip-10-10-143-102:~# enum4linux 10.10.185.8
WARNING: polenum.py is not in your path.  Check that package is installed and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Apr 24 17:3
1:11 2023

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.10.185.8
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ============================================
|    Enumerating Workgroup/Domain on 10.10.185.8    |
 ============================================
[+] Got domain/workgroup name: WORKGROUP
```

---> The machine name can be found in the ‘OS’ information on IP section:

```
 ================================
|    OS information on 10.10.185.8    |
 ================================
Use of uninitialized value $os_info in concatenation (.) or string at /root/Desktop/Tools/Miscellaneou
s/enum4linux.pl line 464.
[+] Got OS info for 10.10.185.8 from smbclient:
[+] Got OS info for 10.10.185.8 from srvinfo:
        POLOSMB        Wk Sv PrQ Unx NT SNT polosmb server (Samba, Ubuntu)
        platform_id     :       500
        os version      :       6.1
        server type     :       0x809a03
```

Here we can see the open shares under ‘Share enumeration’ on <IP> section of the enum4linux output:

```
 ========================================
|    Share Enumeration on 10.10.185.8    |
 ========================================
WARNING: The "syslog" option is deprecated

        Sharename       Type      Comment
        ---------       ----      -------
        netlogon        Disk      Network Logon Service
        profiles        Disk      Users profiles
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (polosmb server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            POLOSMB
```

We'll investigate them all. And the 'profiles' look significant for it can poses user info.

Exploiting SMB

---> We can use smbclient utility to access an SMB share. SMB shares can also have permissions associated with users.

---> smbclient uses the following syntax:

smbclient //<IP>/<share> -U <user> -p <port>

```
smbclient //10.10.185.8/profiles
```

```
root@ip-10-10-143-102:~# smbclient //10.10.185.8/profiles
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Apr 21 12:08:23 2020
  ..                                  D        0  Tue Apr 21 11:49:56 2020
  .cache                             DH        0  Tue Apr 21 12:08:23 2020
  .profile                            H      807  Tue Apr 21 12:08:23 2020
  .sudo_as_admin_successful           H        0  Tue Apr 21 12:08:23 2020
  .bash_logout                        H      220  Tue Apr 21 12:08:23 2020
  .viminfo                            H      947  Tue Apr 21 12:08:23 2020
  Working From Home Information.txt      N      358  Tue Apr 21 12:08:23 2020
  .ssh                               DH        0  Tue Apr 21 12:08:23 2020
  .bashrc                             H     3771  Tue Apr 21 12:08:23 2020
  .gnupg                             DH        0  Tue Apr 21 12:08:23 2020

                12316808 blocks of size 1024. 7583708 blocks available
smb: \>
```

As we said we would look up to the netlogon, profiles, print$ etc. And when we do bingo. We are in. We can know look at the /profiles anonymously.

---> We can read the files with more command.

SNMP

HTTP: Hyper Text Transfer Protocol

---> HTTP is a **connection-oriented** protocol that **uses TCP** as a transport protocol. It operates in port 80.

---> Hypertext is a structured text that uses logical lengths between nodes containing the text.

---> Is an application layer protocol

---> Stateless protocol: No info retained about the other host.

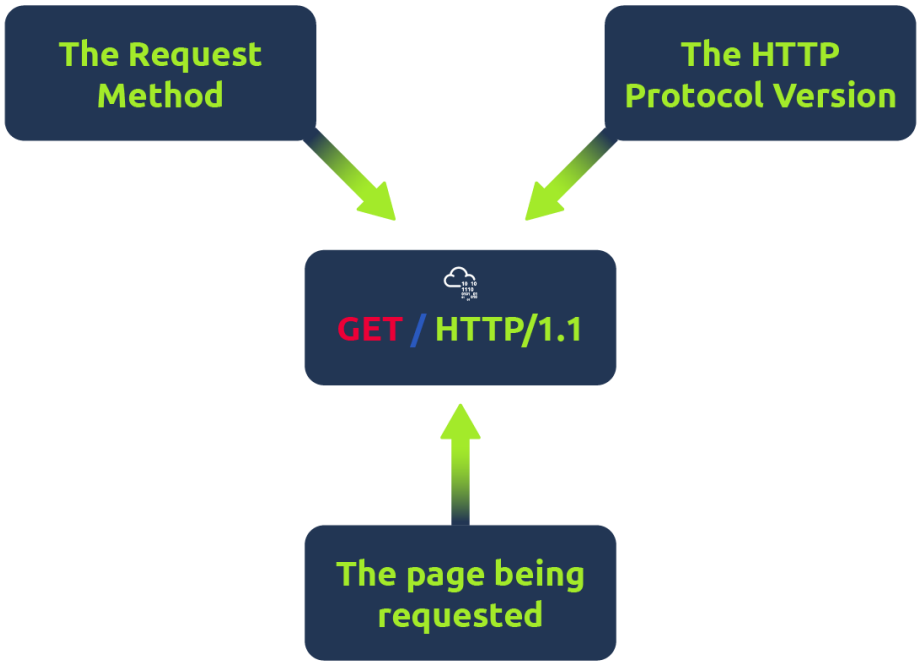
---> Most Used HTTP Request: GET, POST, HEAD, PUT, DELETE, TRACE, OPTIONS etc...

---> The text is stored as plain text. So, anyone seeing the log files can see our text which makes it unsecure.

---> ---> ---> OTW

HTTP Requests

---> Making a request:



All HTTP messages contain the same basic elements.

1.) One or more headers.

2.) Then a blank line

3.) An optional message body

The first line of the HTTP request has 3 elements, separated by spaces.

1.) A verb (action word) indicating the HTTP method. Among these, the most common is GET. The GET method retrieves a resource from the web server.

2.) The requested URL

3.) The HTTP version used.

---> TRYHACKME example:

```
GET / HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Referer: https://tryhackme.com/
```

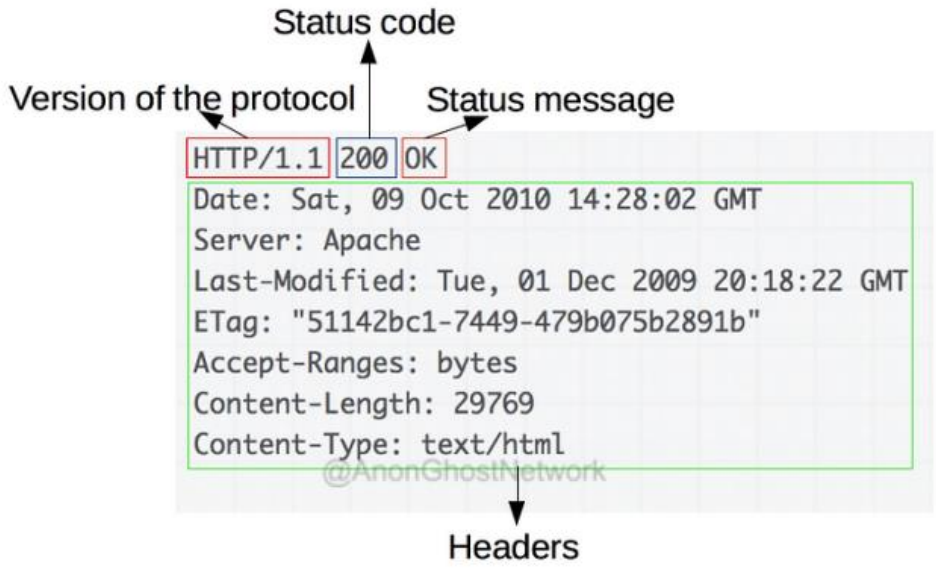
HTTP Responses

A typical HTTP response has 3 items.

1.) The HTTP version

2.) The numeric status code

3.) The text describing the status response.



HTTP Methods

2 most commonly used methods while attacking are GET and POST

The **GET** method is built to retrieve resources

The **POST** method is built to perform actions. Used for submitting data to the web server and potentially create new records

Other methods are:

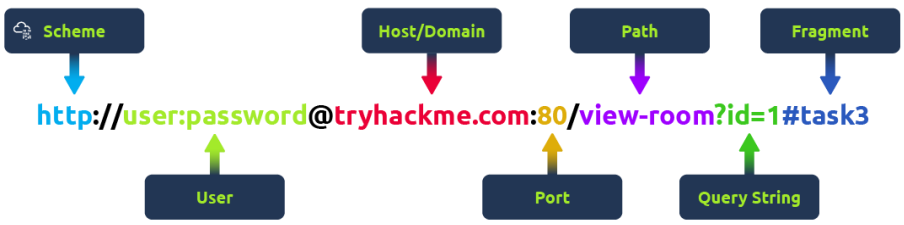
HEAD functions are similar to GET requests. But no message body is returned.

TRACE is used for diagnostic purposes.

OPTIONS asks the server to report HTTP methods are available

PUT attempts to upload a resource to the server to update information, which is contained in the body.

URL's



URL is a unique ID for every web resource. The basic syntax for URL is: **protocol://hostname[:port]/ [/path/] file [?param=value]**

Port number is optional and only necessary if the port is different from the default port used by the protocol specified in the first field. **HTTP = 80**, **HTTPS=443**, **FTP = 21** etc.

HTTP Headers

There are numerous types of Headers in HTTP. Some can be used for both requests and responses, and others are specific to the massage types.

Some of the common header types are:

General Headers

Connection: Tells the other end whether the connection should be closed after HTTP transmission.

Content-Encoding: Specifies the type of encoding

Content-Length: Specifies content length

Content-Type: Specifies the content type.

Transfer-Encoding: Specifies the encoding on the message body.

Request Headers

Accept: Specifies to the server what type of message encoding it'll accept.

Authorization - submits credentials

Cookie - submits cookies to the server

Host - specifies the host name

If-Modified-Since - specifies WHEN the browser last received the resource. If not modified, the server instructs the client to use the cached copy

If-None-Match - specifies entity tag

Origin - specifies the domain where the request originated

Referrer - specifies the URL of the requestor

User-Agent - specifies the browser that generated the request

Request Headers

Host	Servers can host multiple websites so this header tells the server which one you are trying to access.
User-Agent	Tells the server your browser and version, allowing it to format the website so that it is compatible.
Content-Length	Tells the server how much data to expect.
Accept-Encoding	Informs the server what kind of compression methods are supported by the browser.
Cookie	Data to help the server remember information about you.

Response Headers

Set-Cookie	Information that gets stored and sent back to the server.
Cache-Control	Tells the browser how long to store the content in its' cache before requesting again.
Content-Type	This lets the browser know what kind of content to expect.
Content-Encoding	The type of compression method used by the server.

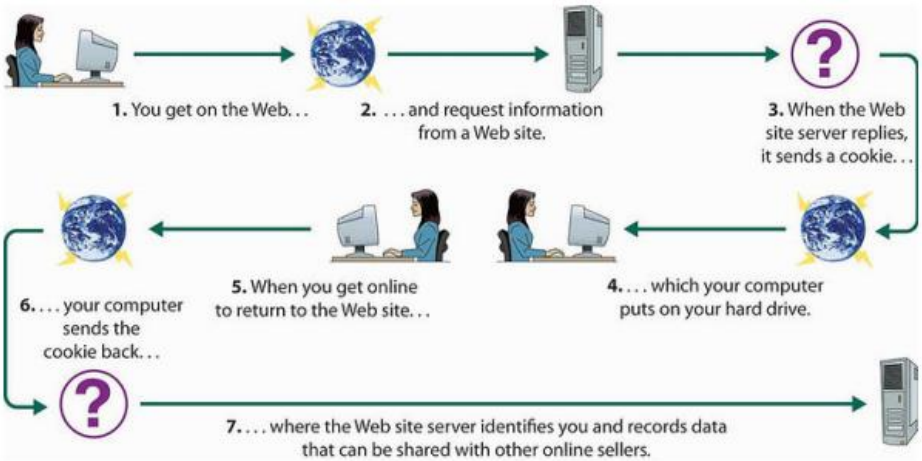
Cookies

Cookies are small piece of data that is stored on our computer. They enable the server to send items of data to the client and client stores this data and resubmits it to the server the next time a request is made to the server.

The server issues a cookie to the client using the SET-COOKIE response header. And in every time a new request is made you send cookie data back to the web server.

SetCookie: Tracking=wdr66gyU34pli89

When the user makes a subsequent request to the server, the cookie is added to the header.



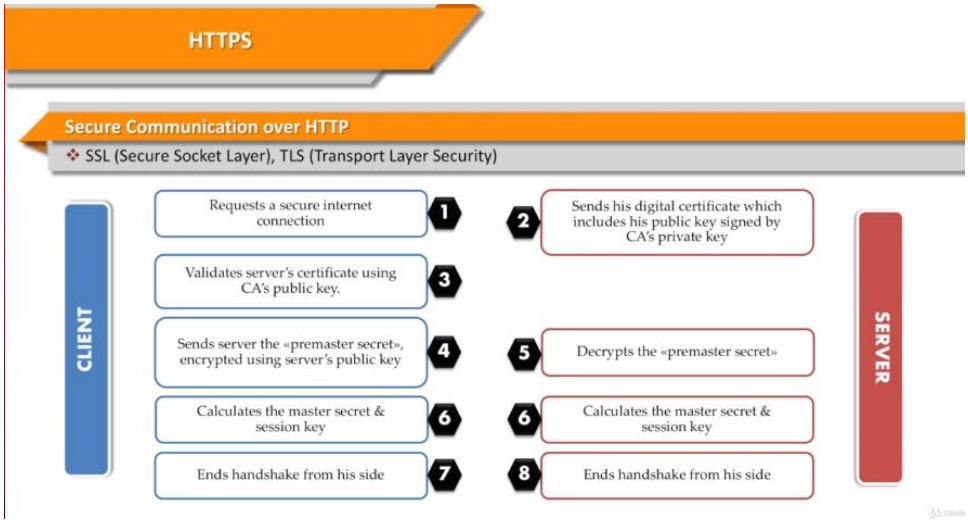
HTTPS

---> What we are calling HTTPS is basically communication with encryption within HTTP.

---> While the security and encryption aspects of HTTPS are implemented in layer 7, the underlying transport protocol, usually TCP, is handled at the Transport Layer.

---> SSL (Secure Socket Layer) and TLS (Transport Layer Security)

---> ---> BUT HOW IT WORKS?



---> ---> ---> OTW

HTTP protocol is transmitted in plain TCP, unencrypted and vulnerable for MitM attacks. HTTPS is almost the same as HTTP, but it is tunneled using SSL.

Önemli Not: Sayfa 178'deki Hacking WEB App Authentication with BurpSuite'e bak.

DHCP (Dynamic Host Configuration Protocol)

---> DHCP: Automatic distributions of IP addresses within a network. it distributes IP addresses automatically. So, you don't need to do it statically. Each time you join a LAN, you are likely to receive a different(dynamic) IP address. But usually in the same range. For instance, 192.168.0.0 - 192.168.255.255.

---> Configures the subnet mask, default gateway and DNS server.

---> The client requests an IP address, the DHCP server assigns an available address.

ADVANTAGES

-> Almost no conflict, easy to manage config.

-> Easier to manage a network

-> Move freely to move between networks.

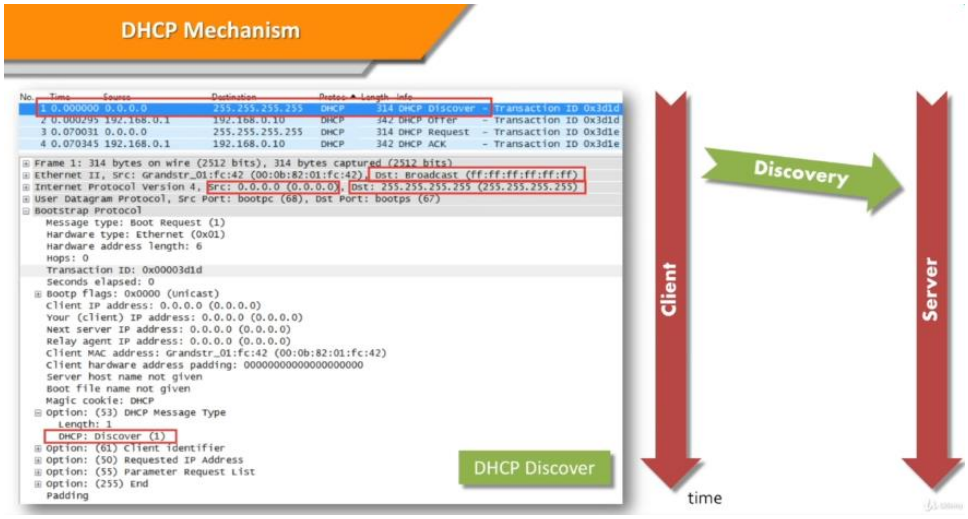
Cyber Security Point of Point

-> The first replying device decides the configuration

-> No authentication for the DHCP server

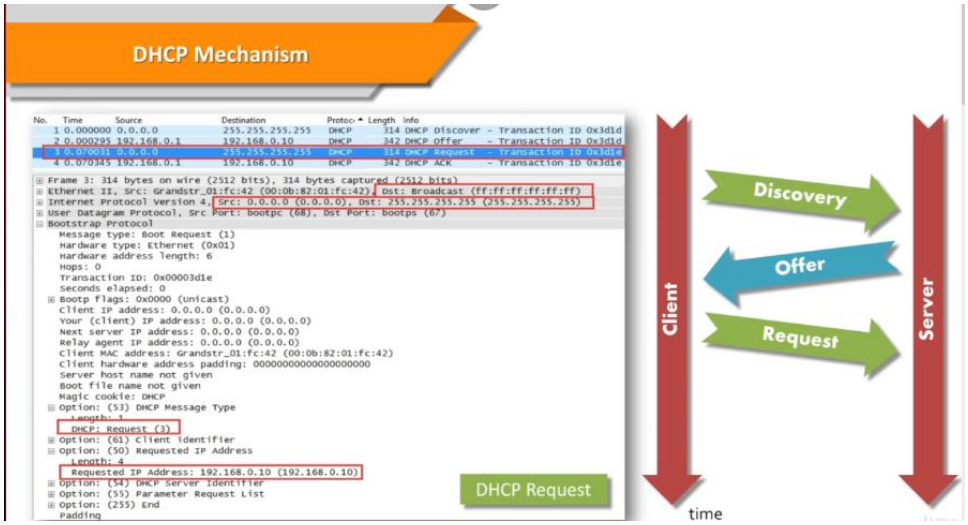
-> No auth. for clients.

---> ---> ---> DHCP Mechanism.

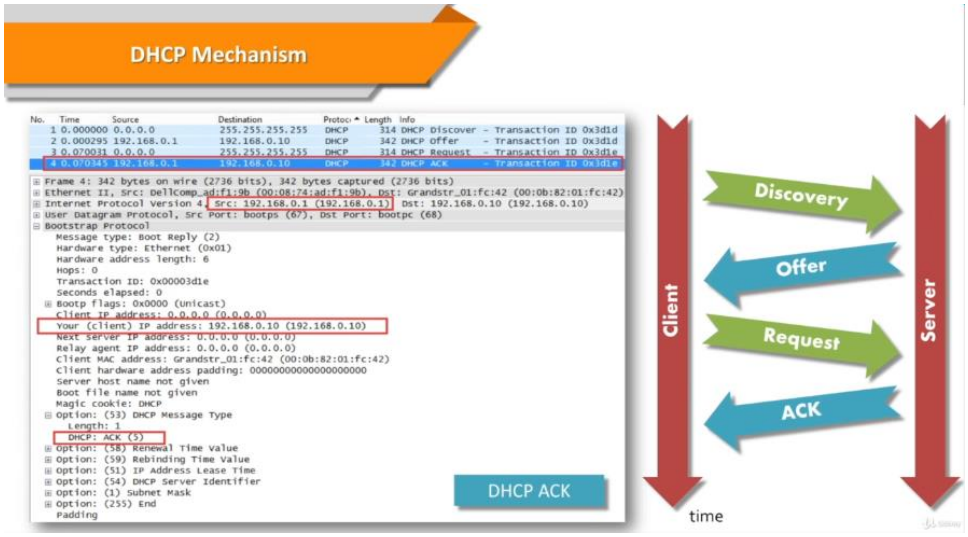


---> When a device connected to a network if an IP isn't assigned to it, it broadcasts to the entire network for and ip address with **DHCP Discover**.

---> The DHCP replies with an IP address that, device can use. **DHCP Offer**



---> Device sends back a message confirming the acceptance of the offer. **DHCP Request**



---> Server sends a reply acknowledging the completement of the process. **DHCP ACK**

File Transfer Protocol (FTP)

FTP provides for file uploading and downloading of files from a remote host running FTP server software. FTP enables us to view the contents of folders on an FTP server and rename and delete files and directories if we have the necessary permission.

FTP is an application layer protocol that uses ports 20 and 21 and sends information unencrypted.

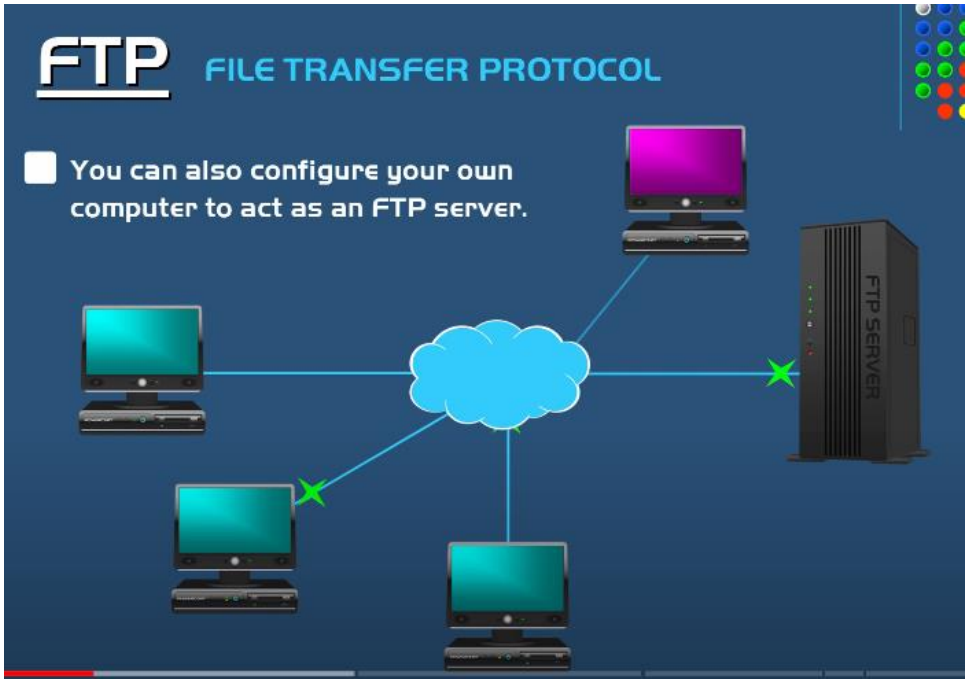
All networks offer FTP server capabilities. FTP assumes that files uploaded and downloaded are straight text (ASCII) files. If not, the transfer mode must be changed to binary.

TABLE 2.3 **Commonly Used FTP Commands**

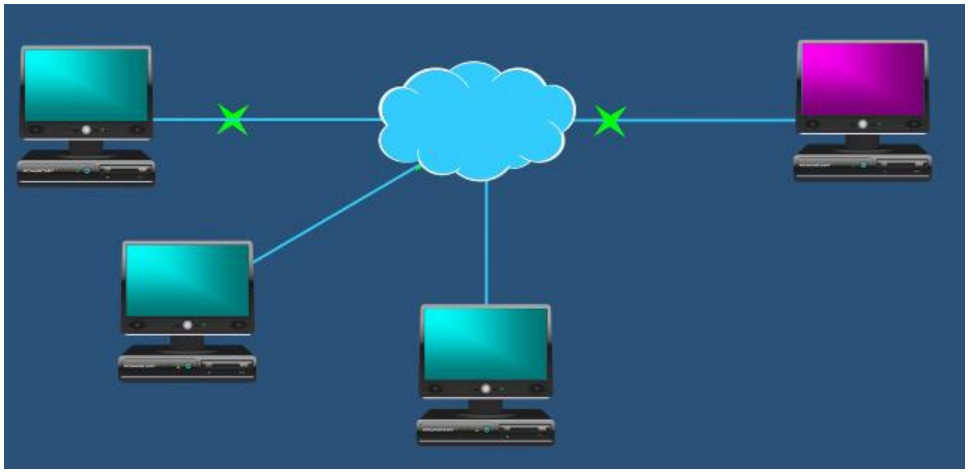
Command	Description
`ls`	Lists the files in the current directory on the remote system
`cd`	Changes the working directory on the remote host
`lcd`	Changes the working directory on the local host
`put`	Uploads a single file to the remote host
`get`	Downloads a single file from the remote host
`mput`	Uploads multiple files to the remote host
`mget`	Downloads multiple files from the remote host
`binary`	Switches transfers into binary mode
`ascii`	Switches transfers into ASCII mode (the default)

FTP From PowerCert Animated Videos

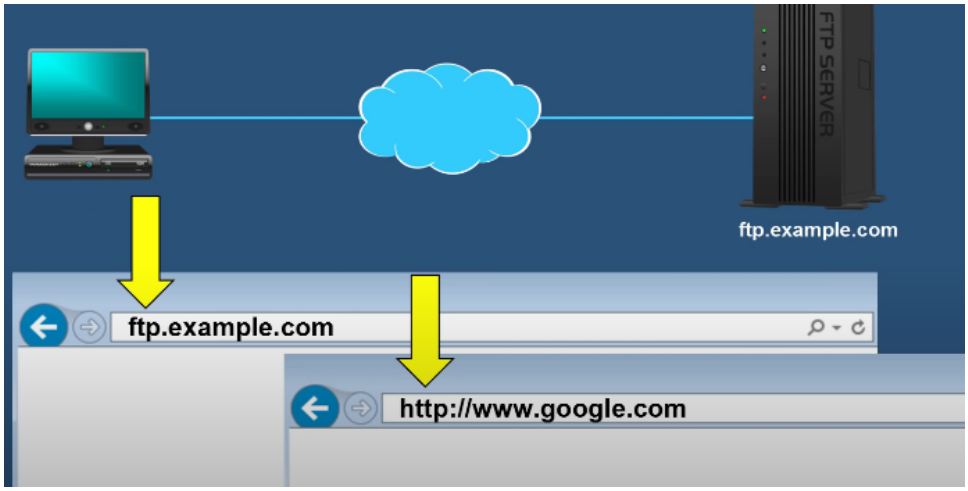
If someone anywhere around the world wants to make their documents available, they need to upload their files to FTP server and other anywhere in the world can reach that FTP server to download it.



But the thing is to do that, a person doesn't have to set an FTP server. Because they can set up their computer as FTP server.

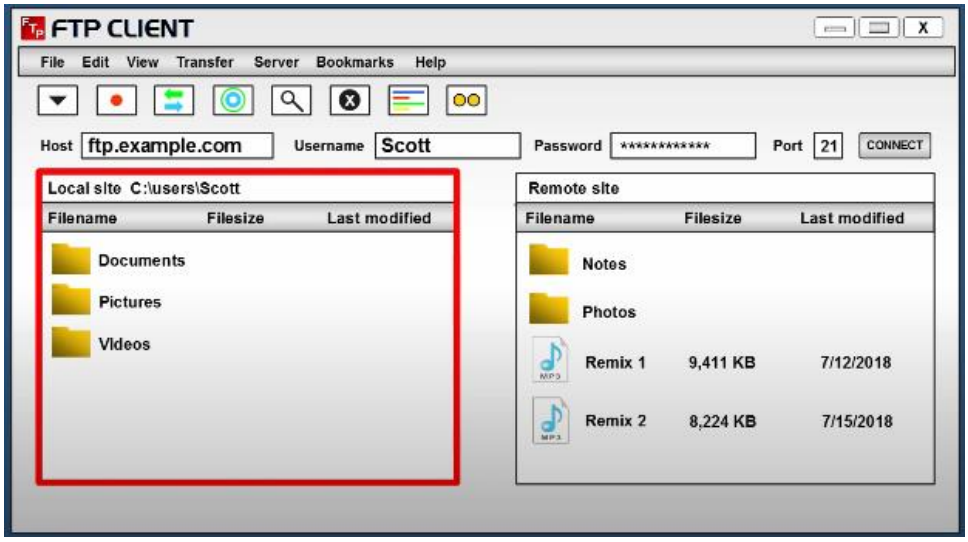


A person can go to the FTP server on internet an download it.



Normally the protocol is HTTP or HTTPS but since we are using ftp, our protocol is FTP.

Sometimes FTP can ask password or you can download anonymously. A popular free FTP client is Filezilla.



On the left we have the files/folders on our computers. And on the right, we have the files/folders on the remote FTP server. You can drop it in both ways.

Another use of FTP is it gives the ability of website designers to upload files to their web servers.

Secure File Transfer Protocol (SFTP)

The main problem with FTP is, it is unsecure. FTP transmits data between sender and receiver in an unencrypted format. So, by using a packet sniffer a hacker can easily get the files.

STFP uses port 22(**SSH uses this port as well**) for secure file transfers.

Note: 2 protocols can't use the same ports at the same time. However, SSH and SFTP can use the port 22 at the same time. The reason being is SFTP is actually a subsystem of SSH and operates over an SSH connection.

Both FTP and SFTP are connection oriented (They use TCP), so they guarantee packet delivery.

Trivial File Transfer Protocol (TFTP)

TFTP is an application layer that **uses UDP**, which is connectionless. For this reason, TFTP is called a connectionless file transfer method.

Not used to transfer files over the internet. Mainly used to transfer files within a local area network.

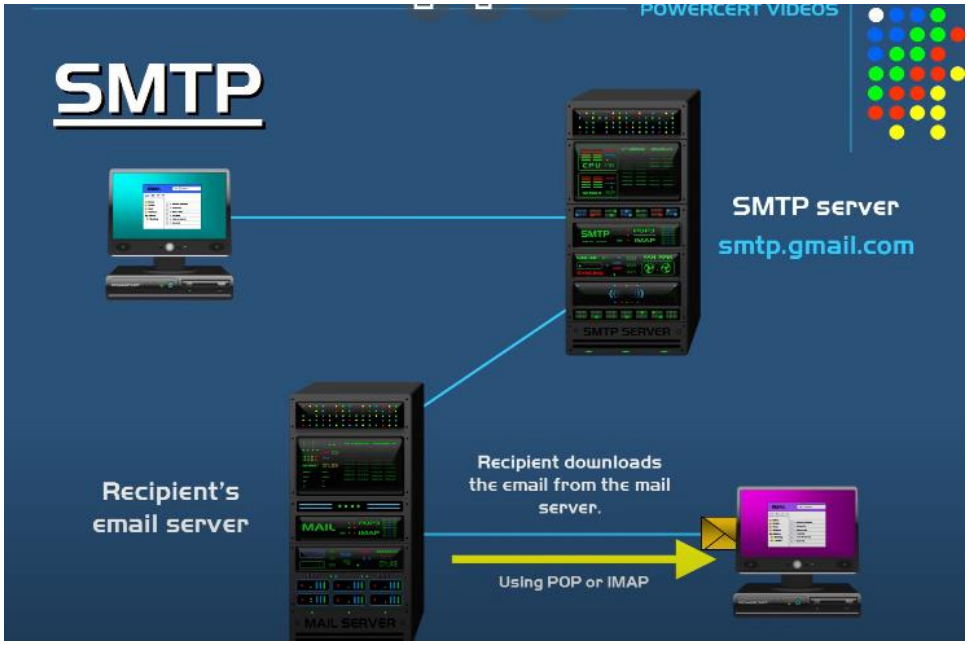
For example, for transferring configuration files and firmware images.

Simple Mail Transfer Protocol (SMTP)

If an attacker can breach your SMTP protocol, they can unmask your VPN communication, like NSA did with the ExtraBacon exploit.

SMTP protocol is an application layer protocol that defines how mail messages are sent between hosts. SMTP uses TCP connections to guarantee error-free delivery of messages. So, in case there is an error (let's say you misspelled it), you get an error message. It requires destination host to be always available, because of this mail systems spool incoming mail so the users can read it later. How the user reads the mail depends on how the client accesses the SMTP server. The default port of SMTP is 25.

SMTP can be used for sending and receiving mail. POP3 and IMAP4 can be used only to receive mail.



You are sending mail to someone, you are using SMTP server. The receiver's server receives the e-mail and receiver downloads the e-mail whenever he wants with POP or IMAP.

Post Office Protocol Version 3/Internet Message Access Protocol Version 4

POP3 and IMAP4 are mechanism for downloading (or pulling), email from a server. You should notice that they can't send e-mail. That is the job of SMTP protocol (Which can both send and receive but can't hold mail for long). POP3 uses port 110 and IMAP4 uses port 143.

The problem with POP3 is that the password used to access a mailbox is transmitted across the network in clear text. This is the area where IMAP4 offers advantage over POP3.

Even though POP3 and IMAP4 can hold mails most of today's systems are using internet-based mails so we can reach our mails anywhere.

Telnet

Used to access UNIX/LINUX systems. Uses port 23 and is insecure. SSH is more secure replacement of Telnet.

SSH

SSH uses port 22 and is a secure alternative for Telnet.

SSH is foundational technology for SFTP.

Note: Both telnet and SSH is used to access remote servers.

Network Time Protocol

---> NTP is used for time synchronization and implemented over UDP port 123.

---> Without keeping up with time, we couldn't keep track of changes to data and applications.

---> Every device has internal clock.

Lightweight Directory Access Protocol (LDAP)

---> LDAP uses port 389, and LDAPS uses 636.

---> Active Directory (AD): Use to provide authentication, group and user management.

---> LDAP: Is a protocol for reading and writing directories over an IP network

---> LDAP uses TCP/IP (TCP/389 and UDP/389)

LDAP is the protocol used to query

DATA ENCAPSULATION

---> Adding protocol info to data as it passes through layers is known as encapsulation. Data is de-encapsulated as it moves up.

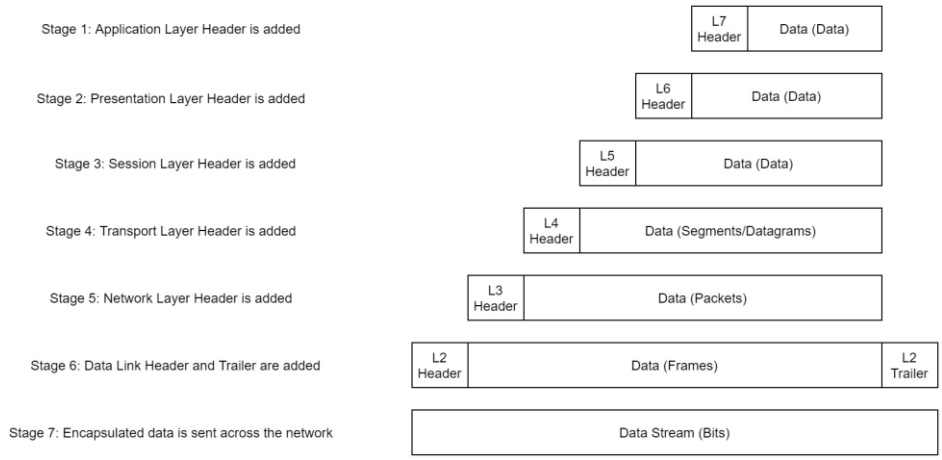
---> A frame is at **layer 2**. In a frame there is no IP address. We can say if we are talking about IP addresses we are talking about packets.

Header	Description
Time to Live	This field sets an expiry timer for the packet to not clog up your network if it never manages to reach a host or escape!
Checksum	This field provides integrity checking for protocols such as TCP/IP. If any data is changed, this value will be different from what was expected and therefore corrupt.
Source Address	The IP address of the device that the packet is being sent **from** so that data knows where to **return to**.
Destination Address	The device's IP address the packet is being sent to so that data knows where to travel next.

TABLE 2.3 **OSI Model Encapsulation/Decapsulation**

OSI Layer	Encapsulation/Decapsulation Function	Representation
Application (Layer 7)	The data is created in the application(s) and passed to/from the Transport layer.	DATA
Presentation (Layer 6)		
Session (Layer 5)		
Transport (Layer 4)	A segment header is added to, or removed from, the data.	SEGMENT HEADER | DATA
Network (Layer 3)	A packet header is added to, or removed from, the data.	PACKET HEADER | SEGMENT HEADER | DATA
Data link (Layer 2)	A frame header is added to, or removed from, the data. A frame trailer is added to, or removed from, the data.	FRAME HEADER | PACKET HEADER | SEGMENT HEADER | DATA | FRAME TRAILER

---> In encapsulation each layer adds headers. For example, header added by network layer includes source and destination IP. Header added by Transport layer includes info specific to the protocol used, the data link layer adds a piece on at the end of the transmission which indicates if the data is corrupted or not.

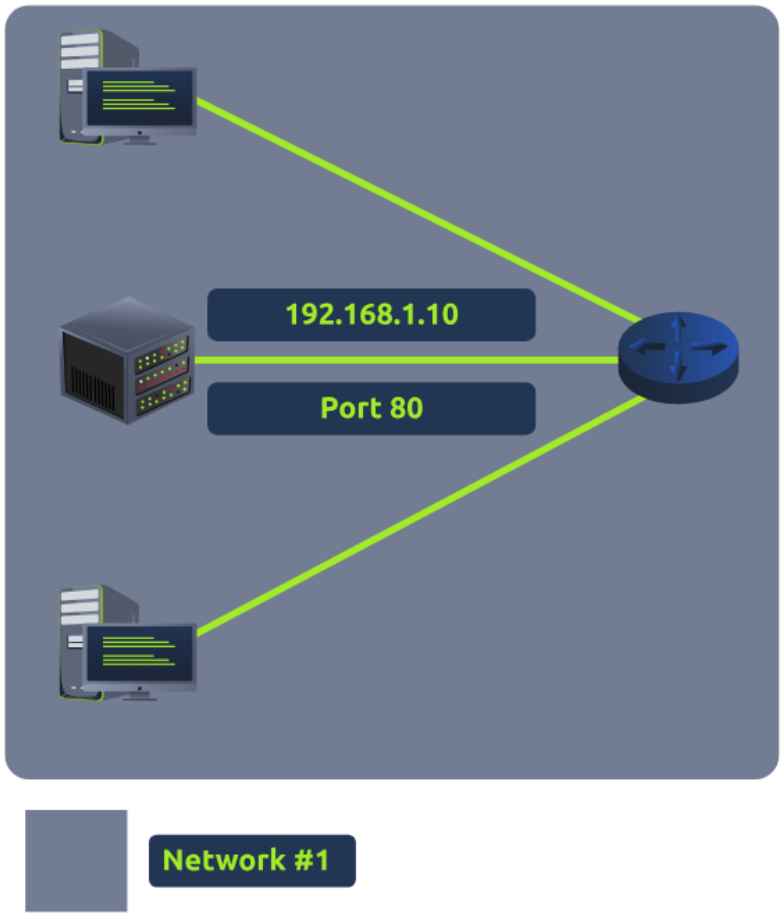


---> In Layer 7,6 and 5 data is referred as data. In layer 4(Transport) data is referred as Segments/datagram. In layer 3(Network) data is referred as Packets. In layer 2(Data) it is referred as Frames. In physical layer it is referred as bits.

Extending Your Network (THM)

FIREWALL

---> Without port forwarding, applications and services would be only available within the same network



---> In the picture above only the 2 other computers will be able to access to the server (which for that is known as intranet) with the IP address 192.168.1.10 which runs in the port 80.

---> If a server needs to be reachable outside it needs port forwarding. Port forwarding is done with **routers**.

---> A **firewall** is a device that can allow a traffic or deny it. They operate in layer 4 and layer 7.

---> Firewalls can be categorized into 2 to 5 categories.

Stateful: **Based upon the entire connection**. This firewall uses entire information from a connection, rather than inspecting individual packet. This type consumes a lot of resources as the decision making is dynamic. If a connection from host is bad, it'll block the entire device.

Stateless: Uses static set of rules to determines whether the **individual packets** are acceptable. They use les resource and are dumber.

VPN

---> VPN allows devices on separate networks to communicate securely by creating a dedicated path between each other over the internet (known as tunnel).

---> Some existing VPN technologies are:

PPP: This technology is used by PPTP to allow for *authentication* and providing *encryption* of data. This Technology isn't capable of routing. It can't leave network itself.

PPTP: Point-to-Point Tunneling Protocol allows data from PPP to *travel and leave* a network.

IPSec: Encrypts the data using the existing IP framework.

Router

---> **Router** routs. It connects different networks and passes data between them. Routing is the label given to the process of data travelling across networks. Routing involves creating a path between networks so that this data can be successfully delivered. It operates in layer 3.

---> **Switch** is a dedicated networking device responsible for providing a means of connecting to multiple devices. It operates in *layer 2 and layer 3*. The sole responsibility of switch is sending *frames to correct devices*.

---> VLAN allows specific devices within a network to be virtually split up.