Network fundamentals

Made by Bob John

I did make these notes how ever, I did not make them all my self. Some are from try hack me.

Different kinds of firewalls consist of

- Firewall appliance: The firewall allows and blocks connections based on a predefined set of rules. It restricts what can enter and what can leave a network.
- Intrusion Detection System (IDS) appliance: An IDS detects system and network intrusions and intrusion attempts. It tries to detect attackers' attempts to break into your network.
- Intrusion Prevention System (IPS) appliance: An IPS blocks detected intrusions and intrusion attempts. It aims to prevent attackers from breaking into your network.
- Virtual Private Network (VPN) concentrator appliance: A VPN ensures that the network traffic cannot be read nor altered by a third party. It protects the confidentiality (secrecy) and integrity of the sent data.

VPN Technology	Description
PPP	This technology is used by PPTP (explained below) to allow for authentication and provide encryption of data. VPNs work by using a private key and public certificate (similar to **SSH**). A private key & certificate must match for you to connect. This technology is not capable of leaving a network by itself (non-routable).
PPTP	The **P**oint-to-**P**oint **T**unneling **P**rotocol (**PPTP**) is the technology that allows the data from PPP to travel and leave a network. PPTP is very easy to set up and is supported by most devices. It is, however, weakly encrypted in comparison to alternatives.
IPSec	Internet Protocol Security (IPsec) encrypts data using the existing **I**nternet **P**rotocol (**IP**) framework. IPSec is difficult to set up in comparison to alternatives; however, if successful, it boasts strong encryption and is also supported on many devices.

Benefit	Description
Allows networks in different geographical locations to be connected.	For example, a business with multiple offices will find VPNs beneficial, as it means that resources like servers/infrastructure can be accessed from another office.
Offers privacy.	VPN technology uses encryption to protect data. This means that it can only be understood between the devices it was being sent from and is destined for, meaning the data isn't vulnerable to sniffing. This encryption is useful in places with public WiFi, where no encryption provided by the network. You can use a VPN to protect your traffic from being viewed by other people.
Offers anonyminity.	Journalists and activists depend upon VPNs to safely report on global issues in countries where freedom of speech is controlled. Usually, your traffic can be viewed by your ISP and other intermediaries and therefore tracked. The level of anonymity a VPN provides is only as much as how other devices on the network respect privacy.. For example, a VPN that logs all of your data/history is essentially the same as not using a VPN in this regard.

Source and destination IP addresses: An *IP address* is a logical address that allows you to communicate over the Internet. One analogy is the postal address; for example, a company needs a valid postal address to send and receive parcels. Think of the IP packet as a mail parcel.

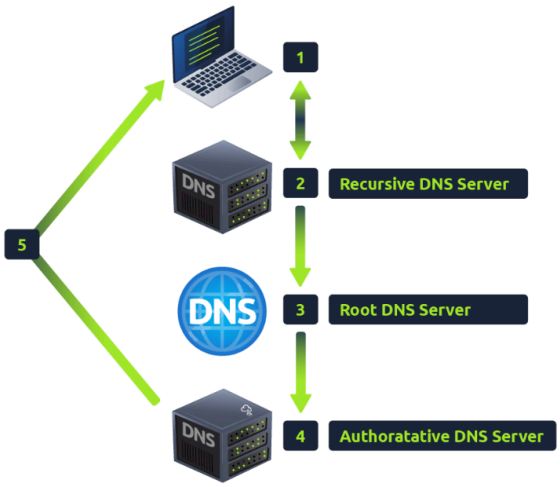
Source and destination port numbers (where applicable): A computer has an IP address; furthermore, each program on the computer needs a *port number* to communicate over the network. Back to our analogy, a port number would be similar to a room number within a company.

DNS

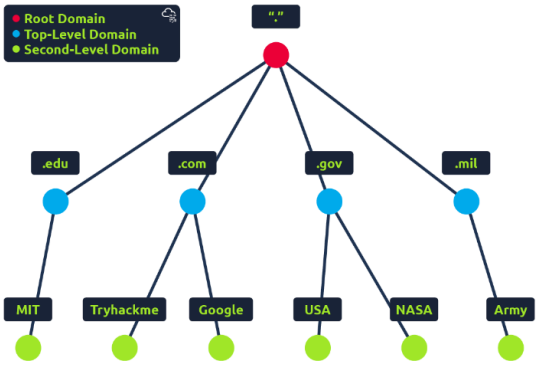
DNS (Domain Name System) provides a simple way for us to communicate with devices on the internet without remembering complex numbers. Much like every house has a unique address for sending mail directly to it, every computer on the internet has its own unique address to communicate with it called an IP address. An IP address looks like the following 104.26.10.229, 4 sets of digits ranging from 0 - 255 separated by a period. When you want to visit a website, it's not exactly convenient to remember this complicated set of numbers, and that's where DNS can help.

What happens when you make a DNS request

1. When you request a domain name, your computer first checks its local cache to see if you've previously looked up the address recently; if not, a request to your Recursive DNS Server will be made.
2. A Recursive DNS Server is usually provided by your ISP, but you can also choose your own. This server also has a local cache of recently looked up domain names. If a result is found locally, this is sent back to your computer, and your request ends here (this is common for popular and heavily requested services such as Google, Facebook, Twitter). If the request cannot be found locally, a journey begins to find the correct answer, starting with the internet's root DNS servers.
3. The root servers act as the DNS backbone of the internet; their job is to redirect you to the correct Top Level Domain Server, depending on your request. If, for example, you request www.tryhackme.com, the root server will recognise the Top Level Domain of .com and refer you to the correct TLD server that deals with .com addresses.
4. The TLD server holds records for where to find the authoritative server to answer the DNS request. The authoritative server is often also known as the nameserver for the domain. For example, the name server for tryhackme.com is kip.ns.cloudflare.com and uma.ns.cloudflare.com. You'll often find multiple nameservers for a domain name to act as a backup in case one goes down.
5. An authoritative DNS server is the server that is responsible for storing the DNS records for a particular domain name and where any updates to your domain name DNS records would be made. Depending on the record type, the DNS record is then sent back to the Recursive DNS Server, where a local copy will be cached for future requests and then relayed back to the original client that made the request. DNS records all come with a TTL (Time To Live) value. This value is a number represented in seconds that the response should be saved for locally until you have to look it up again. Caching saves on having to make a DNS request every time you communicate with a server.



Domain Hierarchy



Root Domain

Everything is under this if it ends with a . then it is under it

TLD Domain's

Top level domain's are your .com .edu. Gov. xyz
There are two types of TLD, gTLD (Generic Top Level) and ccTLD (Country Code Top Level Domain). ccTLD was used for geographical purposes, for example, .ca for sites based in Canada, .co.uk for sites based in the United Kingdom and so on

SLD Domain’s

Second level domains are going to be the name for the website you chose.When registering a domain name, the second-level domain is limited to 63 characters + the TLD and can only use a-z 0-9 and hyphens (cannot start or end with hyphens or have consecutive hyphens).

Subdomain

A subdomain sits on the left-hand side of the Second-Level Domain using a period to separate it; for example, in the name admin.tryhackme.com the admin part is the subdomain. A subdomain name has the same creation restrictions as a Second-Level Domain, being limited to 63 characters and can only use a-z 0-9 and hyphens (cannot start or end with hyphens or have consecutive hyphens). You can use multiple subdomains split with periods to create longer names, such as jupiter.servers.tryhackme.com. But the length must be kept to 253 characters or less. There is no limit to the number of subdomains you can create for your domain name.

When you type something it will go as www.thewebsiteyouwant.com the www stands for (world wide web)

MAC Address stands for media access control

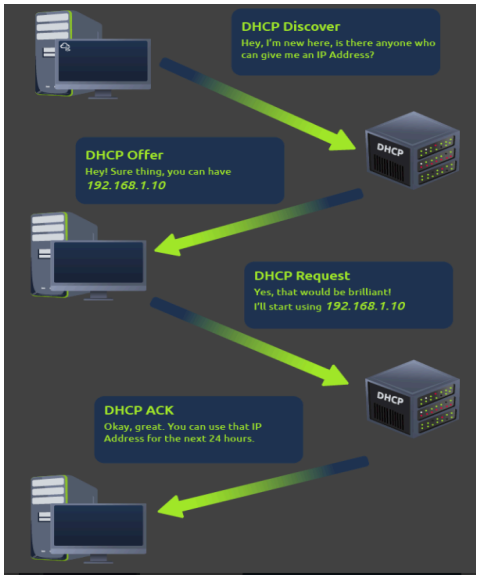
Network Address	This address identifies the start of the actual network and is used to identify a network's existence.	For example, a device with the IP address of 192.168.1.100 will be on the network identified by 192.168.1.0	192.168.1.0
Host Address	An IP address here is used to identify a device on 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

DHCP Protocal

DHCP stands for dynamic host configuration protocol

When a device connects to a network, if it has not already been manually assigned an IP address, it sends out a request (DHCP Discover) to see if any DHCP servers are on the network. The DHCP server then replies back with an IP address the device could use (DHCP Offer). The device then sends a reply confirming it wants the offered IP

Address (DHCP Request), and then lastly, the DHCP server sends a reply acknowledging this has been completed, and the device can start using the IP Address (DHCP ACK).



OSI Model

The OSI model (or Open Systems Interconnection Model) is an absolute fundamental model used in networking. This critical model provides a framework dictating how all networked devices will send, receive and interpret data.

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data link
1. Physical

7. the application layer is the layer in which protocols and rules are in place to determine how the user should interact with data sent or received.

6. Layer 6 of the OSI model is the layer in which standardisation starts to take place.Security features such as data encryption (like HTTPS when visiting a secure site) occur at this layer.

5. Once data has been correctly translated or formatted from the presentation layer (layer 6), the session layer (layer 5) will begin to create a connection to the other computer that the data is destined for. When a connection is established, a session is created. Whilst this connection is active, so is the session.

4. Transport layer 4 transmits data across the network using either **TCP** or **UDP**.

TCP Transmition control protocol

The main aim of tcp is reliability

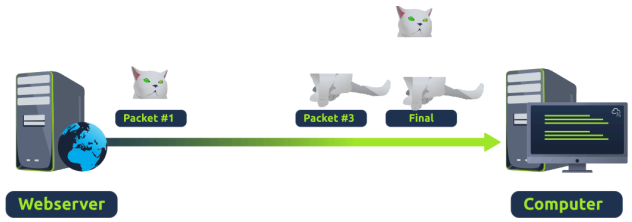
This protocol reserves a constant connection between the two devices for the amount of time it takes for the data to be sent and received. TCP incorporates error checking into its design. Error checking is how TCP can guarantee that data sent from the packets in the session layer (layer 5) has then been received and reassembled in the same order.

Advantages of TCP	Disadvantages of TCP
Guarantees the accuracy of data.	Requires a reliable connection between the two devices. If one small chunk of data is not received, then the entire chunk of data cannot be used.
Capable of synchronising two devices to prevent each other from being flooded with data.	A slow connection can bottleneck another device as the connection will be reserved on the receiving computer the whole time.
Performs a lot more processes for reliability.	TCP is significantly slower than UDP because more work has to be done by the devices using this protocol.

TCP is used for situations such as file sharing, internet browsing or sending an email. This usage is because these services require the data to be accurate and complete

UDP user datagram protocal

any data that gets sent via UDP is sent to the computer whether it gets there or not. There is no synchronisation between the two devices or guarantee; just hope for the best, and fingers crossed.



Advantages of UDP	Disadvantages of UDP
UDP is much faster than TCP.	UDP doesn't care if the data is received.
UDP leaves the application layer (user software) to decide if there is any control over how quickly packets are sent.	It is quite flexible to software developers in this sense.
UDP does not reserve a continuous connection on a device as TCP does.	This means that unstable connections result in a terrible experience for the user.

3. OSPF (Open Shortest Path First) and RIP (Routing Information Protocol). The factors that decide what route is taken is decided by the following:

- What path is the shortest? I.e. has the least amount of devices that the packet needs to travel across.
- What path is the most reliable? I.e. have packets been lost on that path before?
- Which path has the faster physical connection? I.e. is one path using a copper connection (slower) or a fibre (considerably faster)?

At this layer, everything is dealt with via IP addresses such as 192.168.1.100. Devices such as routers capable of delivering packets using IP addresses are known as Layer 3 devices — because they are capable of working at the third layer of the OSI model.

2. The data link layer focuses on the physical addressing of the transmission. It receives a packet from the network layer (including the IP address for the remote computer) and adds in the physical MAC (Media Access Control) address of the receiving endpoint. Inside every network-enabled computer is a Network Interface Card (NIC) which comes with a unique MAC address to identify it.

1. Physical connected cables that transfer bytes of data.

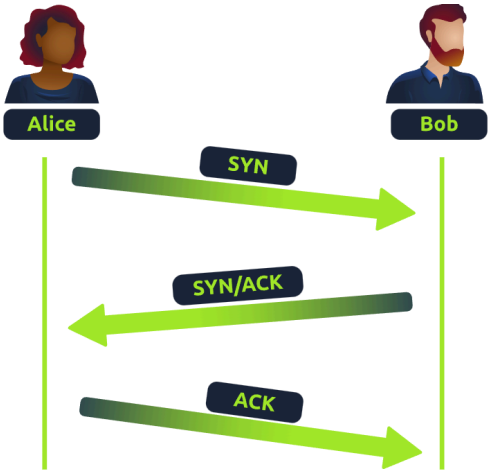
3 way hand shake

TCP packets contain various sections of information known as headers that are added from encapsulation

Header	Description
Source Port	This value is the port opened by the sender to send the TCP packet from. This value is chosen randomly (out of the ports from 0-65535 that aren't already in use at the time).
Destination Port	This value is the port number that an application or service is running on the remote host (the one receiving data); for example, a webserver running on port 80. Unlike the source port, this value is not chosen at random.
Source IP	This is the IP address of the device that is sending the packet.
Destination IP	This is the IP address of the device that the packet is destined for.
Sequence Number	When a connection occurs, the first piece of data transmitted is given a random number. We'll explain this more in-depth further on.
Acknowledgement Number	After a piece of data has been given a sequence number, the number for the next piece of data will have the sequence number + 1. We'll also explain this more in-depth further on.
Checksum	This value is what gives TCP integrity. A mathematical calculation is made where the output is remembered. When the receiving device performs the mathematical calculation, the data must be corrupt if the output is different from what was sent.
Data	This header is where the data, i.e. bytes of a file that is being transmitted, is stored.
Flag	This header determines how the packet should be handled by either device during the handshake process. Specific flags will determine specific behaviours, which is what we'll come on to explain below.

Step	Message	Description
1	SYN	A SYN message is the initial packet sent by a client during the handshake. This packet is used to initiate a connection and synchronise the two devices together (we'll explain this further later on).
2	SYN/ACK	This packet is sent by the receiving device (server) to acknowledge the synchronisation attempt from the client.
3	ACK	The acknowledgement packet can be used by either the client or server to acknowledge that a series of messages/packets have been successfully received.
4	DATA	Once a connection has been established, data (such as bytes of a file) is sent via the "DATA" message.
5	FIN	This packet is used to *cleanly (properly)* close the connection after it has been complete.
#	RST	This packet abruptly ends all communication. This is the last resort and indicates there was some problem during the process. For example, if the service or application is not working correctly, or the system has faults such as low resources.

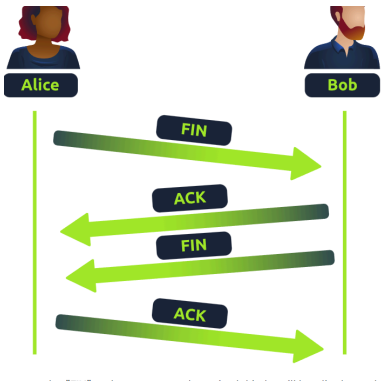
TCP establishing a conection



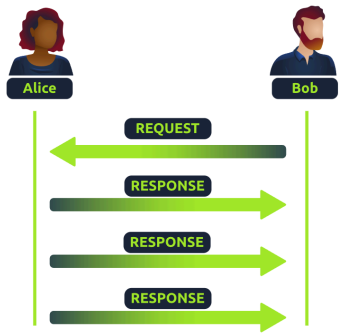
Any sent data is given a random number sequence and is reconstructed using this number sequence and incrementing by 1. Both computers must agree on the same number sequence for data to be sent in the correct order. This order is agreed upon during three steps:

1. SYN - Client: Here's my Initial Sequence Number(ISN) to SYNchronise with (0)
2. SYN/ACK - Server: Here's my Initial Sequence Number (ISN) to SYNchronise with (5,000), and I ACKnowledge your initial number sequence (0)
3. ACK - Client: I ACKnowledge your Initial Sequence Number (ISN) of (5,000), here is some data that is my ISN+1 (0 + 1)

Device	Initial Number Sequence (ISN)	Final Number Sequence
Client (Sender)	0	0 + 1 = 1
Client (Sender)	1	1 + 1 = 2
Client (Sender)	2	2 + 1 = 3



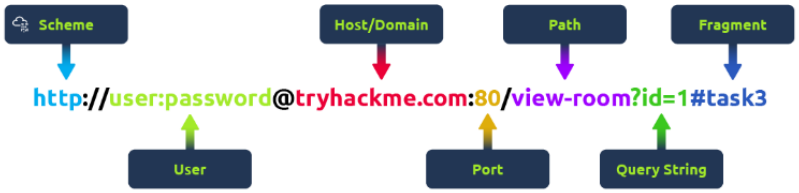
Header	Description
Time to Live (TTL)	This field sets an expiry timer for the packet, so it doesn't clog up your network if it never manages to reach a host or escape!
Source Address	The IP address of the device that the packet is being sent from, so that data knows where to return to.
Destination Address	The device's IP address the packet is being sent to so that data knows where to travel next.
Source Port	This value is the port that is opened by the sender to send the TCP packet from. This value is randomly chosen (out of the ports from 0-65535 that aren't already in use at the time).
Destination Port	This value is the port number that an application or service is running on the remote host (the one receiving the data); for example, a webserver running on port 80. Unlike the source port, this value is not chosen at random.
Data	This header is where data, i.e. bytes of a file that is being transmitted, is stored.



Protocol	Port Number	Description
File Transfer Protocol (FTP)	21	This protocol is used by a file-sharing application built on a client-server model, meaning you can download files from a central location.
Secure Shell (SSH)	22	This protocol is used to securely login to systems via a text-based interface for management.
HyperText Transfer Protocol (HTTP)	80	This protocol powers the World Wide Web (WWW)! Your browser uses this to download text, images and videos of web pages.
HyperText Transfer Protocol Secure (HTTPS)	443	This protocol does the exact same as above; however, securely using encryption.
Server Message Block (SMB)	445	This protocol is similar to the File Transfer Protocol (FTP); however, as well as files, SMB allows you to share devices like printers.
Remote Desktop Protocol (RDP)	3389	This protocol is a secure means of logging in to a system using a visual desktop interface (as opposed to the text-based limitations of the SSH protocol).

HTTP

URL? (Uniform Resource Locator)
A URL is predominantly an instruction on how to access a resource on the internet.



scheme: This instructs on what protocol to use for accessing the resource such as HTTP, HTTPS, FTP (File Transfer Protocol).

User: Some services require authentication to log in, you can put a username and password into the URL to log in.

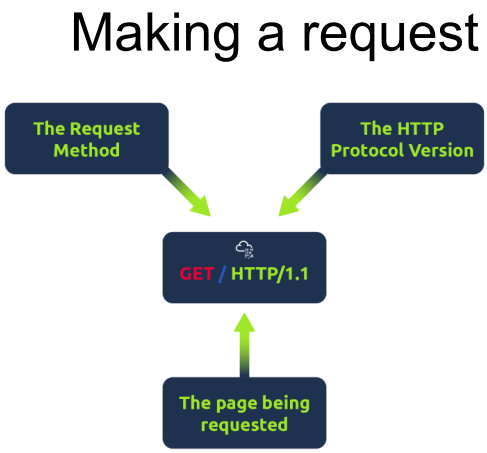
Host: The domain name or IP address of the server you wish to access.

Port: The Port that you are going to connect to, usually 80 for HTTP and 443 for HTTPS, but this can be hosted on any port between 1 - 65535.

Path: The file name or location of the resource you are trying to access.

Query String: Extra bits of information that can be sent to the requested path. For example, /blog?id=1 woulSd tell the blog path that you wish to receive the blog article with the id of 1.

Fragment: This is a reference to a location on the actual page requested. This is commonly used for pages with long content and can have a certain part of the page directly linked to it, so it is viewable to the user as soon as they access the page.



This is extremely basic for more your going to want to get more data from the webserver.

Example:

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

Line 1: This request is sending the GET method (more on this in the HTTP Methods task), request the home page with / and telling the web server we are using HTTP protocol version 1.1.

Line 2: We tell the web server we want the website tryhackme.com

Line 3: We tell the web server we are using the Firefox version 87 Browser

Line 4: We are telling the web server that the web page that referred us to this one is https://tryhackme.com

Line 5: HTTP requests always end with a blank line to inform the web server that the request has finished.

Response

A very basic and simple response of this would be something like

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Fri, 09 Apr 2021 13:34:03 GMT
Content-Type: text/html
Content-Length: 98

<html>
<head>
    <title>TryHackMe</title>
</head>
<body>
    Welcome To TryHackMe.com
</body>
</html>
```

Line 1: HTTP 1.1 is the version of the HTTP protocol the server is using and then followed by the HTTP Status Code in this case "200 Ok" which tells us the request has completed successfully.

Line 2: This tells us the web server software and version number.

Line 3: The current date, time and timezone of the web server.

Line 4: The Content-Type header tells the client what sort of information is going to be sent, such as HTML, images, videos, pdf, XML.

Line 5: Content-Length tells the client how long the response is, this way we can confirm no data is missing.

Line 6: HTTP response contains a blank line to confirm the end of the HTTP response.

Lines 7-14: The information that has been requested, in this instance the homepage.

HTTP Methods

GET Request

This is used for getting information from a web server.

POST Request

This is used for submitting data to the web server and potentially creating new records

PUT Request

This is used for submitting data to a web server to update information

DELETE Request

This is used for deleting information/records from a web server.