Acknowledgment

This material is taken from /based on the forum “TryHackMe” (https://tryhackme.com/paths)

Pre Security Learning Path

This learning path teaches the pre-requisite technical knowledge to get started in cyber security. To attack or defend any technology, you have to first learn how this technology works.

This path comprises of the following five modules:

- ✓ Module 1: Introduction To Cyber Security
  - Intro to Offensive Security [Access TryHackMe for detail]
  - Intro to Defensive Security
  - Careers in Cyber
- ✓ Module 2: Network Fundamentals
  - What is Networking? [Access TryHackMe for detail]
  - Intro to LAN
  - OSI Model
  - Packets & Frames
- ✓ Module 3: How the web Works
  - DNS in Detail
  - HTTP in Detail
  - How Websites work
  - Putting it altogether
- ✓ Module 4: Linux Fundamentals
  - Linux Fundamentals Part 1
  - Linux Fundamentals Part 2
  - Linux Fundamentals Part 3
- ✓ Module 5: Windows Fundamentals
  - Windows Fundamentals Part 1
  - Windows Fundamentals Part 2
  - Windows Fundamentals Part 3

Module 1: Introduction To Cyber Security

Room 1: Intro to Offensive Security [Accessible on TryHackMe]

Room 2: Intro to Defensive Security

Task 1: Intro to Defensive Security

- ✓ The process of **protecting** an organization's network and computer systems by **analyzing** and **securing** any potential digital threats;
- ✓ somewhat the opposite of offensive security, as it is concerned with two main tasks:
  - Preventing intrusions from occurring
  - Detecting intrusions when they occur and responding properly
- ✓ **Investigating** infected computers or devices to understand, how it was hacked, tracking down cybercriminals, or monitoring infrastructure for malicious activity.
- ✓ Blue teams are part of the defensive security landscape.

Some of the tasks that are related to defensive security include:

- ✓ User cyber security awareness: Training users about cyber security helps protect against various attacks that target their systems.
- ✓ Documenting and managing assets: We need to know the types of systems and devices that we have to manage and protect properly.
- ✓ Updating and patching systems: Ensuring that computers, servers, and network devices are correctly updated and patched against any known vulnerability (weakness).
- ✓ Setting up preventative security devices: firewall and intrusion prevention systems (IPS) are critical components of preventative security. Firewalls control what network traffic can go inside and what can leave the system or network. IPS blocks any network traffic that matches present rules and attack signatures.
- ✓ Setting up logging and monitoring devices: Without proper logging and monitoring of the network, it won't be possible to detect malicious activities and intrusions. If a new unauthorized device appears on our network, we should be able to know.

Task 2: Areas of Defensive Security

Two main topics related to defensive security:

- ✓ Security Operations Center (SOC), where we cover Threat Intelligence
- ✓ Digital Forensics and Incident Response (DFIR), where we also cover Malware Analysis

Security Operations Center (SOC)

A *Security Operations Center* (SOC) is a team of cyber security professionals that monitors the network and its systems to detect malicious cyber security events. Some of the main areas of interest for a SOC are:

- **Vulnerabilities:** Whenever a system vulnerability (weakness) is discovered, it is essential to fix it by installing a proper update or patch. When a fix is not available, the necessary

measures should be taken to prevent an attacker from exploiting it. Although remediating vulnerabilities is of vital interest to a SOC, it is not necessarily assigned to them.
- **Policy violations:** We can think of a security policy as a set of rules required for the protection of the network and systems. For example, it might be a policy violation if users start uploading confidential company data to an online storage service.
- **Unauthorized activity:** Consider the case where a user's login name and password are stolen, and the attacker uses them to log into the network. A SOC needs to detect such an event and block it as soon as possible before further damage is done.
- **Network intrusions:** No matter how good your security is, there is always a chance for an intrusion. An intrusion can occur when a user clicks on a malicious link or when an attacker exploits a public server. Either way, when an intrusion occurs, we must detect it as soon as possible to prevent further damage.

Security operations cover various tasks to ensure protection; one such task is threat intelligence.

Threat Intelligence

Intelligence refers to information you gather about actual and potential enemies. A *threat* is any action that can disrupt or adversely affect a system. **Threat intelligence aims to gather information to help the company better prepare against potential adversaries.** The purpose would be to achieve a *threat-informed defense*. Different companies have different adversaries. Some adversaries might seek to steal customer data from a mobile operator; however, other adversaries are interested in halting the production in a petroleum refinery. Example adversaries include a nation-state cyber army working for political reasons and a ransomware group acting for financial purposes. Based on the company (target), we can expect adversaries.

Intelligence needs data. Data has to be collected, processed, and analyzed. Data collection is done from local sources such as network logs and public sources such as forums. Processing of data aims to arrange them into a format suitable for analysis. The analysis phase seeks to find more information about the attackers and their motives; moreover, it aims to create a list of recommendations and actionable steps.

Learning about your adversaries allows you to know their tactics, techniques, and procedures. As a result of threat intelligence, we identify the threat actor (adversary), predict their activity, and consequently, we will be able to mitigate their attacks and prepare a response strategy.

Digital Forensics and Incident Response (DFIR)

This section is about Digital Forensics and Incident Response (DFIR), and we will cover:

- Digital Forensics
- Incident Response
- Malware Analysis

Digital Forensics

Forensics is the application of science to investigate crimes and establish facts. With the use and spread of digital systems, such as computers and smartphones, a new branch of forensics was born to investigate related crimes: computer forensics, which later evolved into, *digital forensics*.

In defensive security, the focus of digital forensics shifts to analyzing evidence of an attack and its perpetrators and other areas such as intellectual property theft, cyber espionage, and possession of unauthorized content. Consequently, digital forensics will focus on different areas such as:

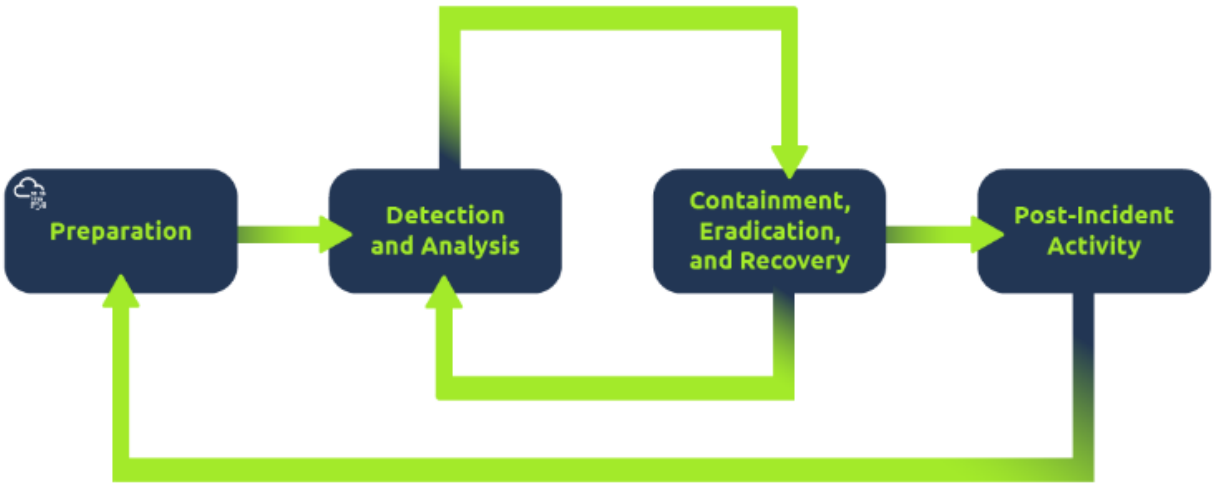
- File System: Analyzing a digital forensics image (low-level copy) of a system's storage reveals much information, such as installed programs, created files, partially overwritten files, and deleted files.
- System memory: If the attacker is running their malicious program in memory without saving it to the disk, taking a forensic image (low-level copy) of the system memory is the best way to analyze its contents and learn about the attack.
- System logs: Each client and server computer maintains different log files about what is happening. Log files provide plenty of information about what happened on a system. Some traces will be left even if the attacker tries to clear their traces.
- Network logs: Logs of the network packets that have traversed a network would help answer more questions about whether an attack is occurring and what it entails.

Incident Response

An *incident* usually refers to a data breach or cyber attack; however, in some cases, it can be something less critical, such as a misconfiguration, an intrusion attempt, or a policy violation. Examples of a cyber attack include an attacker making our network or systems inaccessible, defacing (changing) the public website, and data breach (stealing company data). How would you *respond* to a cyber attack? Incident response specifies the methodology that should be followed to handle such a case. The aim is to reduce damage and recover in the shortest time possible. Ideally, you would develop a plan ready for incident response.

The four major phases of the incident response process are:

1. **Preparation:** This requires a team trained and ready to handle incidents. Ideally, various measures are put in place to prevent incidents from happening in the first place.
2. **Detection and Analysis:** The team has the necessary resources to detect any incident; moreover, it is essential to further analyze any detected incident to learn about its severity.
3. **Containment, Eradication, and Recovery:** Once an incident is detected, it is crucial to stop it from affecting other systems, eliminate it, and recover the affected systems. For instance, when we notice that a system is infected with a computer virus, we would like to stop (contain) the virus from spreading to other systems, clean (eradicate) the virus, and ensure proper system recovery.
4. **Post-Incident Activity:** After successful recovery, a report is produced, and the learned lesson is shared to prevent similar future incidents.



Malware Analysis

Malware stands for malicious software. *Software* refers to programs, documents, and files that you can save on a disk or send over the network. Malware includes many types, such as:

- **Virus** is a piece of code (part of a program) that attaches itself to a program. It is designed to spread from one computer to another; moreover, it works by altering, overwriting, and deleting files once it infects a computer. The result ranges from the computer becoming slow to unusable.
- **Trojan Horse** is a program that shows one desirable function but hides a malicious function underneath. For example, a victim might download a video player from a shady website that gives the attacker complete control over their system.
- Ransomware is a malicious program that encrypts the user's files. Encryption makes the files unreadable without knowing the encryption password. The attacker offers the user the encryption password if the user is willing to pay a "ransom."

Malware analysis aims to learn about such malicious programs using various means:

1. Static analysis works by inspecting the malicious program without running it. Usually, this requires solid knowledge of assembly language (processor's instruction set, i.e., computer's fundamental instructions).
2. Dynamic analysis works by running the malware in a controlled environment and monitoring its activities. It lets you observe how the malware behaves when running.

Task 3: Practical Example of Defensive Security

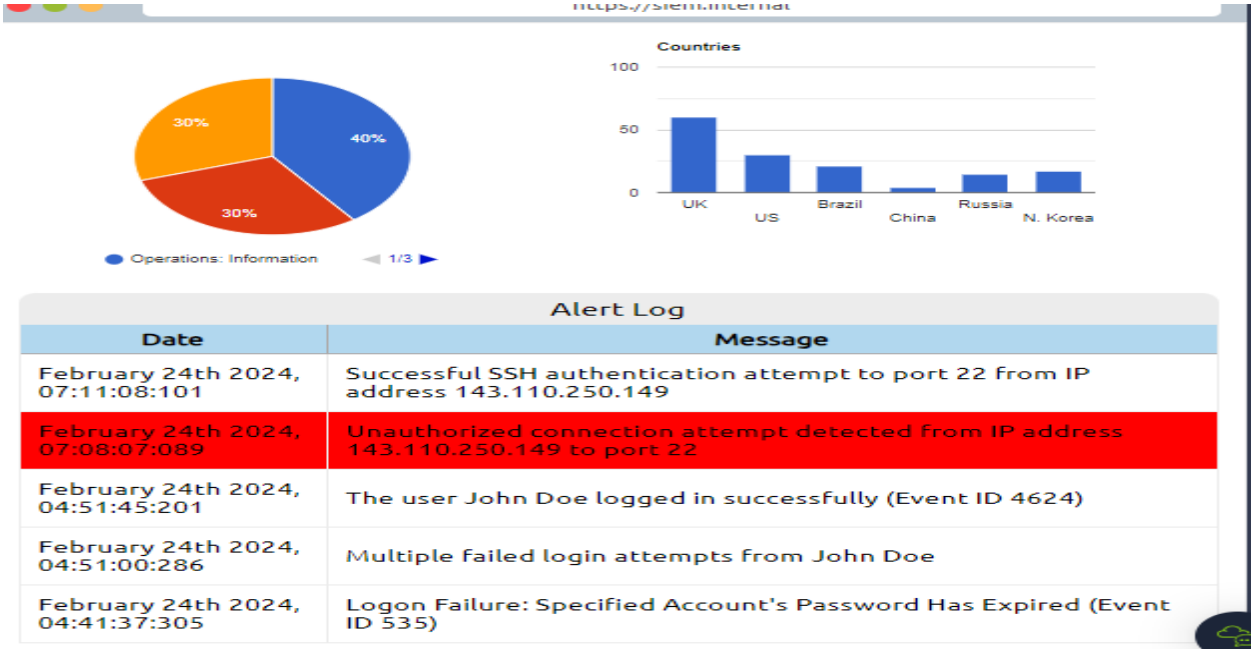
You are part of a *Security Operations Center* (SOC) responsible for protecting a bank. This bank's SOC uses a *Security Information and Event Management* (SIEM) system. A SIEM gathers security-related information and events from various sources and presents them via one system. For instance, you would be notified if there is a failed login attempt or a login attempt from an unexpected geographic location. Moreover, with the advent of machine learning, a SIEM might

detect unusual behavior, such as a user logging in at 3 AM when he usually logs in only during work hours.

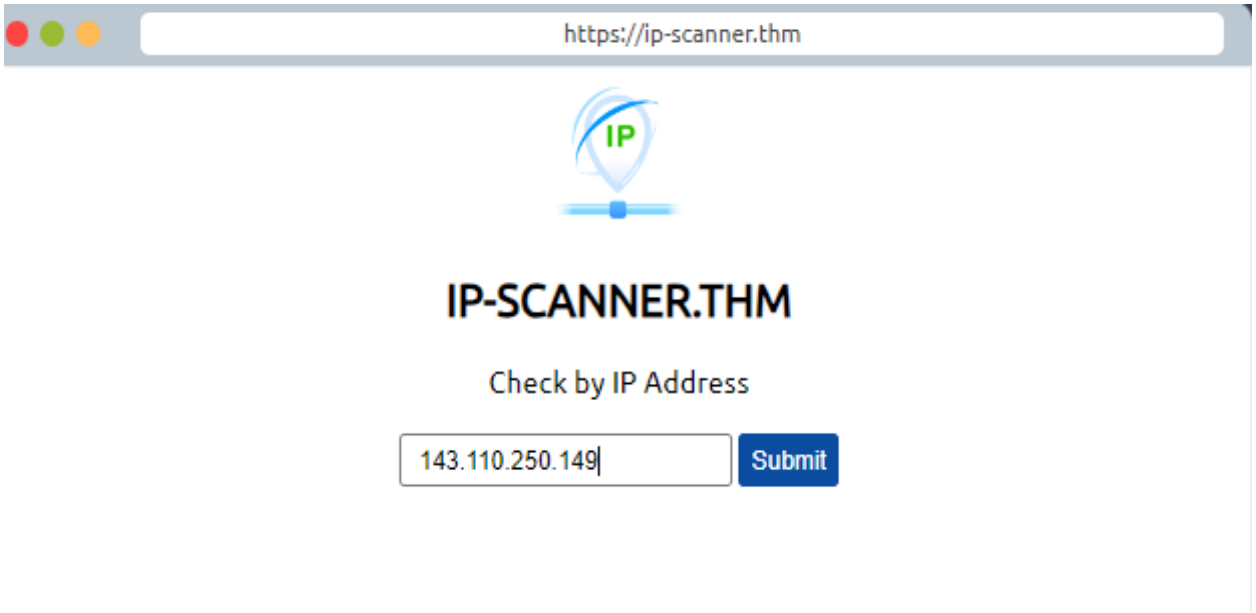
In this exercise, we will interact with a SIEM to monitor the different events on our network and systems in real-time. Some of the events are typical and harmless; others might require further intervention from us. Find the event flagged in red, take note of it, and click on it for further inspection.

Next, we want to learn more about the suspicious activity or event. The suspicious event might have been triggered by an event, such as a local user, a local computer, or a remote IP address. To send and receive postal mail, you need a physical address; similarly, you need an IP address to send and receive data over the Internet. An IP address is a logical address that allows you to communicate over the Internet. We inspect the cause of the trigger to confirm whether the event is indeed malicious. If it is malicious, we need to take due action, such as reporting to someone else in the SOC and blocking the IP address.

Inspect the alerts in your SIEM dashboard. Find the malicious IP address from the alerts, make a note of it, and then click on the alert to proceed.



There are websites on the Internet that allow you to check the reputation of an IP address to see whether it's malicious or suspicious.



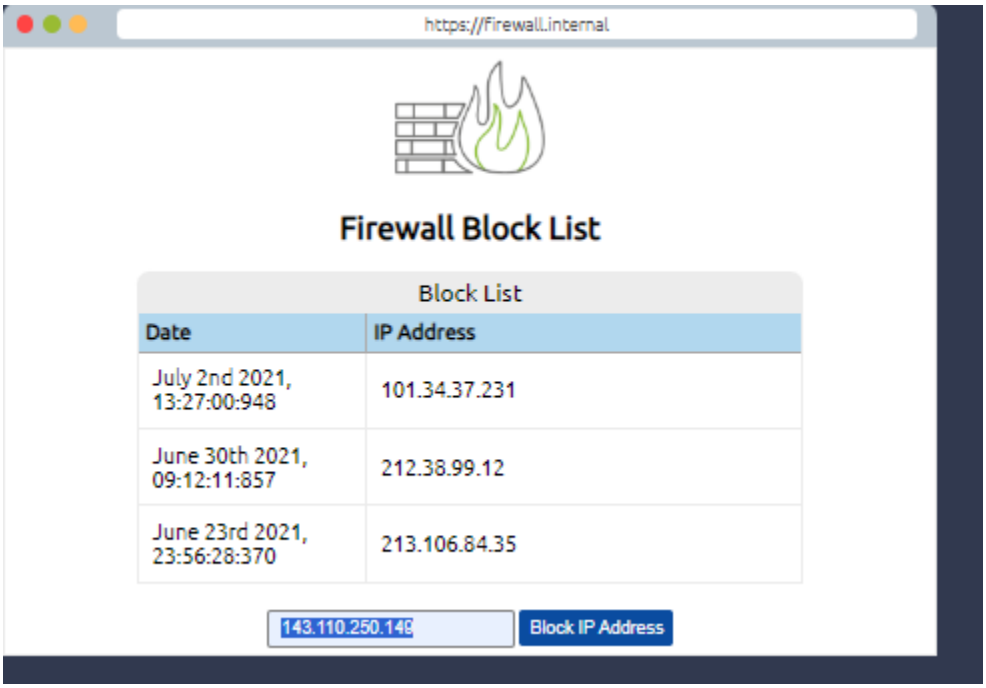
There are many open-source databases out there, like AbuseIPDB, and Cisco Talos Intelligence, where you can perform a reputation and location check for the IP address. Most security analysts use these tools to aid them with alert investigations. You can also make the Internet safer by reporting the malicious IPs, for example, on AbuseIPDB.

Now that we know the IP address is malicious, we need to escalate it to a staff member!

We shouldn't worry too much if it was a failed authentication attempt, but you probably noticed the successful authentication attempt from the malicious IP address. Let's declare a small incident event and escalate it. There is some great staff working at the company, but you wouldn't want to escalate this to the wrong person who is not in charge of your team or department.

You will report it to SOC team lead.

You got the permission to block the malicious IP address, and now you can proceed and implement the block rule. Block the malicious IP address on the firewall and find out what message they left for you.



Room 3: Careers in Cyber

Task 1: Introduction

Cyber security careers are becoming more in demand and offer high salaries. There are many different jobs within the security industry, from offensive pentesting (hacking machines and reporting on vulnerabilities) to defensive security (defending against and investigating cyberattacks).

Why get a career in cyber:

- High Pay - jobs in security have high starting salaries
- Exciting - work can include legally hacking systems or defending against cyber attacks
- Be in demand - there are over 3.5 million unfilled cyber positions

Task 2: Security Analyst

“Responsible for maintaining the security of an organisation's data”

Security analysts are integral to constructing security measures across organisations to protect the company from attacks. Analysts explore and evaluate company networks to uncover actionable data and recommendations for engineers to develop preventative measures. This job

role requires working with various stakeholders to gain an understanding of security requirements and the security landscape.

Responsibilities

- Working with various stakeholders to analyse the cyber security throughout the company
- Compile ongoing reports about the safety of networks, documenting security issues and measures taken in response
- Develop security plans, incorporating research on new attack tools and trends, and measures needed across teams to maintain data security.

Task 3: Security Engineer

"Design, monitor and maintain security controls, networks, and systems to help prevent cyberattacks"

Security engineers develop and implement security solutions using threats and vulnerability data - often sourced from members of the security workforce. Security engineers work across circumventing a breadth of attacks, including web application attacks, network threats, and evolving trends and tactics. The ultimate goal is to retain and adopt security measures to mitigate the risk of attack and data loss.

Responsibilities

- Testing and screening security measures across software
- Monitor networks and reports to update systems and mitigate vulnerabilities
- Identify and implement systems needed for optimal security

Task 4: Incident Responder

"Identifies and mitigates attacks whilst an attackers operations are still unfolding"

Incident responders respond productively and efficiently to security breaches. Responsibilities include creating plans, policies, and protocols for organisations to enact during and following incidents. This is often a highly pressurised position with assessments and responses required in real-time, as attacks are unfolding. Incident response metrics include MTTD, MTTA, and MTTR - the meantime to detect, acknowledge, and recover (from attacks.) The aim is to achieve a swift and effective response, retain financial standing and avoid negative breach implications. Ultimately, incident responders protect the company's data, reputation, and financial standing from cyber attacks.

Responsibilities

- Developing and adopting a thorough, actionable incident response plan

- Maintaining strong security best practices and supporting incident response measures
- Post-incident reporting and preparation for future attacks, considering learnings and adaptations to take from incidents

Task 5: Digital Forensic Examiner

"Responsible for using digital forensics to investigate incidents and crimes"

If you like to play detective, this might be the perfect job. If you are working as part of a law-enforcement department, you would be focused on collecting and analysing evidence to help solve crimes: charging the guilty and exonerating the innocent. On the other hand, if your work falls under defending a company's network, you will be using your forensic skills to analyse incidents, such as policy violations.

Responsibilities

- Collect digital evidence while observing legal procedures
- Analyse digital evidence to find answers related to the case
- Document your findings and report on the case

Task 6: Malware Analyst

"Analyses all types of malware to learn more about how they work and what they do"

A malware analyst's work involves analysing suspicious programs, discovering what they do and writing reports about their findings. A malware analyst is sometimes called a reverse-engineer as their core task revolves around converting compiled programs from machine language to readable code, usually in a low-level language. This work requires the malware analyst to have a strong programming background, especially in low-level languages such as assembly language and C language. The ultimate goal is to learn about all the activities that a malicious program carries out, find out how to detect it and report it.

Responsibilities

- Carry out static analysis of malicious programs, which entails reverse-engineering
- Conduct dynamic analysis of malware samples by observing their activities in a controlled environment
- Document and report all the findings

Task 7: Penetration Tester

"Responsible for testing technology products for security loopholes"

You may see penetration testing referred to as pentesting and ethical hacking. A penetration tester's job role is to test the security of the systems and software within a company - this is achieved through attempts to uncover flaws and vulnerabilities through systemised hacking. Penetration testers exploit these vulnerabilities to evaluate the risk in each instance. The company can then take these insights to rectify issues to prevent a real-world cyberattack.

Responsibilities

- Conduct tests on computer systems, networks, and web-based applications
- Perform security assessments, audits, and analyse policies
- Evaluate and report on insights, recommending actions for attack prevention

Task 8: Red Teamer

"Plays the role of an adversary, attacking an organisation and providing feedback from an enemies perspective"

Red teamers share similarities to penetration testers, with a more targeted job role. Penetration testers look to uncover many vulnerabilities across systems to keep cyber-defence in good standing, whilst red teamers are enacted to test the company's detection and response capabilities. This job role requires imitating cyber criminals' actions, emulating malicious attacks, retaining access, and avoiding detection. Red team assessments can run for up to a month, typically by a team external to the company. They are often best suited to organisations with mature security programs in place.

Responsibilities

- Emulate the role of a threat actor to uncover exploitable vulnerabilities, maintain access and avoid detection
- Assess organisations' security controls, threat intelligence, and incident response procedures
- Evaluate and report on insights, with actionable data for companies to avoid real-world instances

Module 2: Network Fundamentals

Room 1: What is Networking? [Accessible on TryHackMe]

Room 2: Intro to LAN

Task 1: Introduction to LAN Topologies

Local Area Network (LAN) Topologies

Over the years, there has been experimentation and implementation of various network designs. In reference to networking, when we refer to the term "topology", we are actually referring to the design or look of the network at hand. Let's discuss the advantages and disadvantages of these topologies below.

Star Topology

The main premise of a star topology is that devices are individually connected via a central networking device such as a switch or hub. This topology is the most commonly found today because of its reliability and scalability - despite the cost.

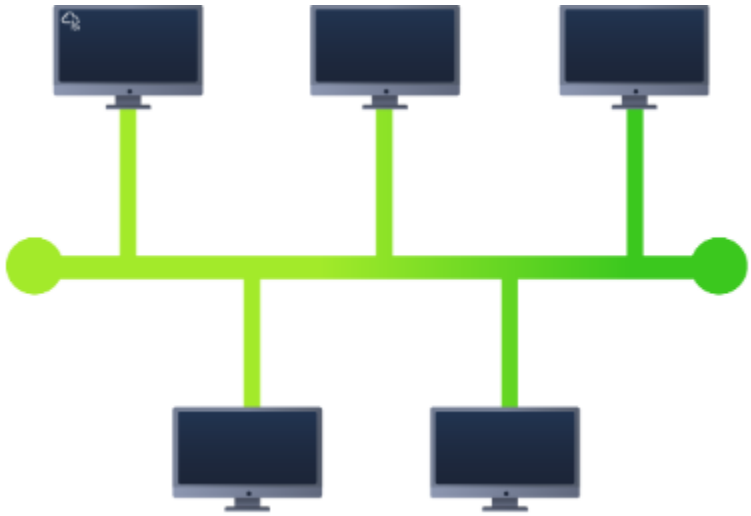
Any information sent to a device in this topology is sent via the central device to which it connects. Let's explore some of these advantages and disadvantages of this topology below:

Because more cabling and the purchase of dedicated networking equipment is required for this topology, it is more expensive than any of the other topologies. However, despite the added cost, this does provide some significant advantages. For example, this topology is much more scalable in nature, which means that it is very easy to add more devices as the demand for the network increases.

Unfortunately, the more the network scales, the more maintenance is required to keep the network functional. This increased dependence on maintenance can also make troubleshooting faults much harder. Furthermore, the star topology is still prone to failure - albeit reduced. For example, if the centralised hardware that connects devices fails, these devices will no longer be able to send or receive data. Thankfully, these centralised hardware devices are often robust

Bus Topology

This type of connection relies upon a single connection which is known as a backbone cable. This type of topology is similar to the leaf off of a tree in the sense that devices (leaves) stem from where the branches are on this cable.



Because all data destined for each device travels along the same cable, it is very quickly prone to becoming slow and bottlenecked if devices within the topology are simultaneously requesting data. This bottleneck also results in very difficult troubleshooting because it quickly becomes difficult to identify which device is experiencing issues with data all travelling along the same route.

However, with this said, bus topologies are one of the easier and more cost-efficient topologies to set up because of their expenses, such as cabling or dedicated networking equipment used to connect these devices.

Lastly, another disadvantage of the bus topology is that there is little redundancy in place in case of failures. This disadvantage is because there is a single point of failure along the backbone cable. If this cable were to break, devices can no longer receive or transmit data along the bus.

Ring Topology

The ring topology (also known as token topology) boasts some similarities. Devices such as computers are connected directly to each other to form a loop, meaning that there is little cabling required and less dependence on dedicated hardware such as within a star topology.

A ring topology works by sending data across the loop until it reaches the destined device, using other devices along the loop to forward the data. Interestingly, a device will only send received data from another device in this topology if it does not have any to send itself. If the device happens to have data to send, it will send its own data first before sending data from another device.

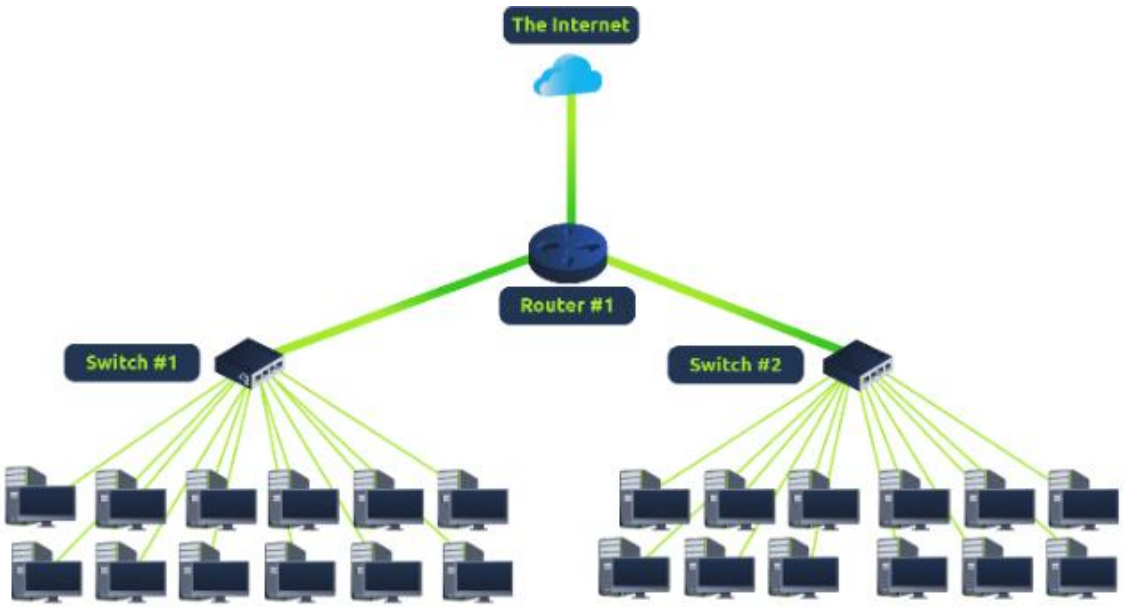
Because there is only one direction for data to travel across this topology, it is fairly easy to troubleshoot any faults that arise. However, this is a double-edged sword because it isn't an efficient way of data travelling across a network, as it may have to visit many multiple devices first before reaching the intended device.

Lastly, ring topologies are less prone to bottlenecks, such as within a bus topology, as large amounts of traffic are not travelling across the network at any one time. The design of this topology does, however, mean that a fault such as cut cable, or broken device will result in the entire networking breaking.

What is a Switch?

Switches are dedicated devices within a network that are designed to aggregate multiple other devices such as computers, printers, or any other networking-capable device using ethernet. These various devices plug into a switch's port. Switches are usually found in larger networks such as businesses, schools, or similar-sized networks, where there are many devices to connect to the network. Switches can connect a large number of devices by having ports of 4, 8, 16, 24, 32, and 64 for devices to plug into.

Switches are much more efficient than their lesser counterpart (hubs/repeaters). Switches keep track of what device is connected to which port. This way, when they receive a packet, instead of repeating that packet to every port like a hub would do, it just sends it to the intended target, thus reducing network traffic.

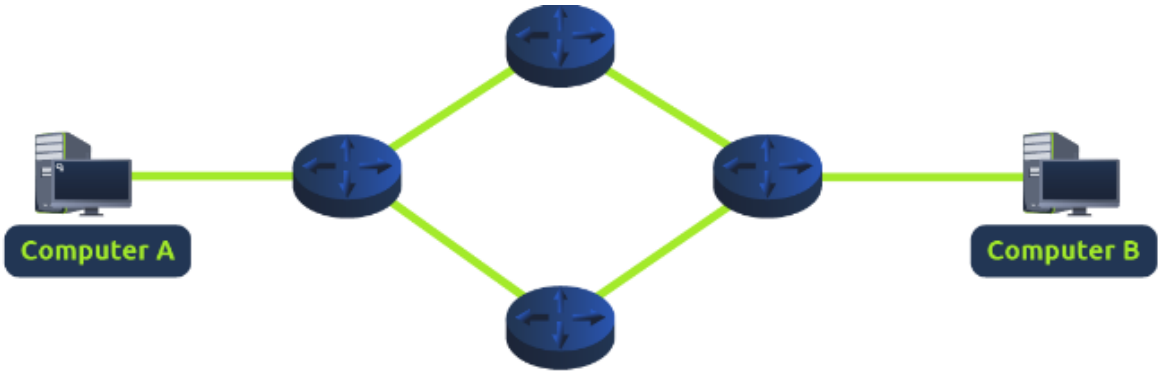


Both Switches and Routers can be connected to one another. The ability to do this increases the redundancy (the reliability) of a network by adding multiple paths for data to take. If one path goes down, another can be used. Whilst this may reduce the overall performance of a network because packets have to take longer to travel, there is no downtime -- a small price to pay considering the alternative.

What is a Router?

It's a router's job to connect networks and pass data between them. It does this by using routing (hence the name router!).

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.



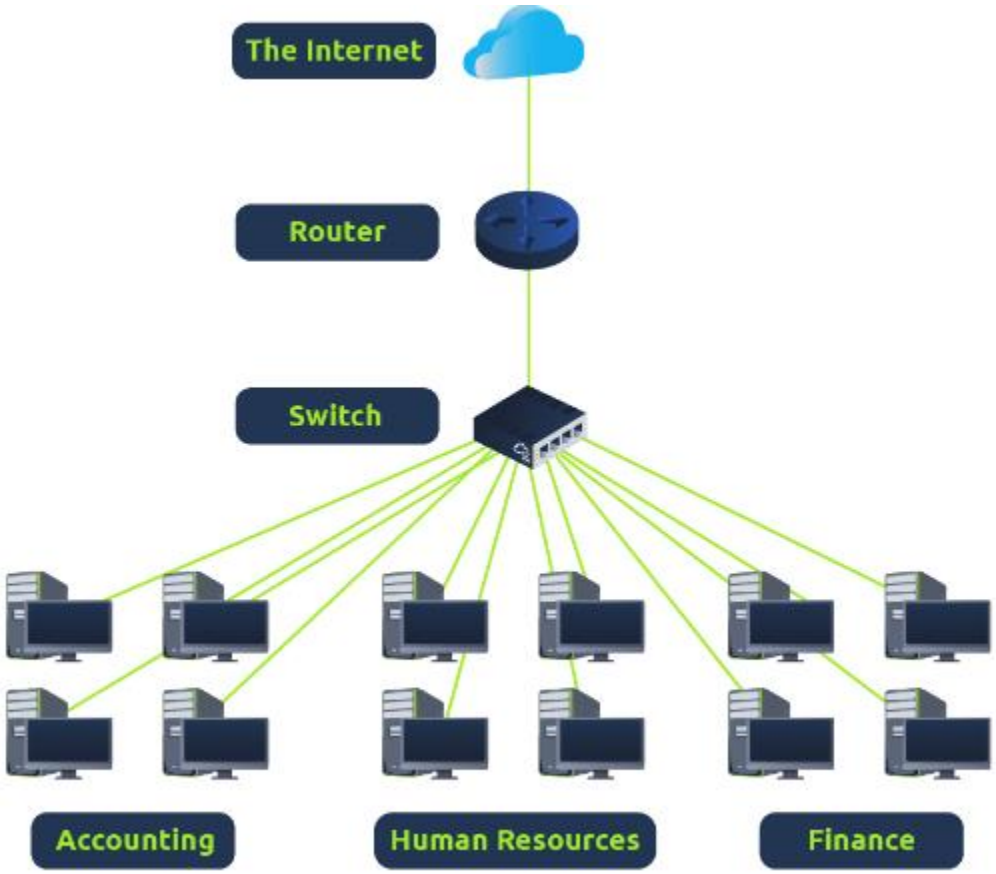
Routing is useful when devices are connected by many paths, such as in the example diagram below.

Task 2: A premier on subnetting

Networks can be found in all shapes and sizes - ranging from small to large. Subnetting is the term given to splitting up a network into smaller, miniature networks within itself. Think of it as slicing up a cake for your friends. There's only a certain amount of cake to go around, but everybody wants a piece. Subnetting is you deciding who gets what slice & reserving such a slice of this metaphorical cake.

Take a business, for example; You will have different departments such as:

- Accounting
- Finance
- Human Resources



Whilst you know where to send information in real life to the correct department, networks need to know as well. Network administrators use subnetting to categorise and assign specific parts of a network to reflect this.

Subnetting is achieved by splitting up the number of hosts that can fit within the network, represented by a number called a subnet mask. Let's refer back to our diagram from the first room in this modul

recall, an IP address is made up of four sections called octets. The same goes for a subnet mask which is also represented as a number of four bytes (32 bits), ranging from 0 to 255 (0-255).

Subnets use IP addresses in three different ways:

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

- Identify the network address
- Identify the host address
- Identify the default gateway

Let's split these three up to understand their purposes into the table below:

Now, in small networks such as at home, you will be on one subnet as there is an unlikely chance that you need more than 254 devices connected at one time.

However, places such as businesses and offices will have much more of these devices (PCs, printers, cameras and sensors), where subnetting takes place.

Subnetting provides a range of benefits, including:

- Efficiency
- Security
- Full control

We'll come on to explore exactly how subnetting provides these benefits at a later date; however, for now, all we need to understand is the security element to it. Let's take the typical café on the street. This cafe will have two networks:

1. One for employees, cash registers, and other devices for the facility
2. One for the general public to use as a hotspot

Subnetting allows you to separate these two use cases from each other whilst having the benefits of a connection to larger networks such as the Internet.

Task 3: The ARP Protocol

Recalling from our previous tasks that devices can have two identifiers: A MAC address and an IP address, the **A**ddress **R**esolution **P**rotocol or **ARP** for short, is the technology that is responsible for allowing devices to identify themselves on a network.

Simply, the ARP protocol allows a device to associate its MAC address with an IP address on the network. Each device on a network will keep a log of the MAC addresses associated with other devices.

When devices wish to communicate with another, they will send a broadcast to the entire network searching for the specific device. Devices can use the ARP protocol to find the MAC address (and therefore the physical identifier) of a device for communication.

How does ARP Work?

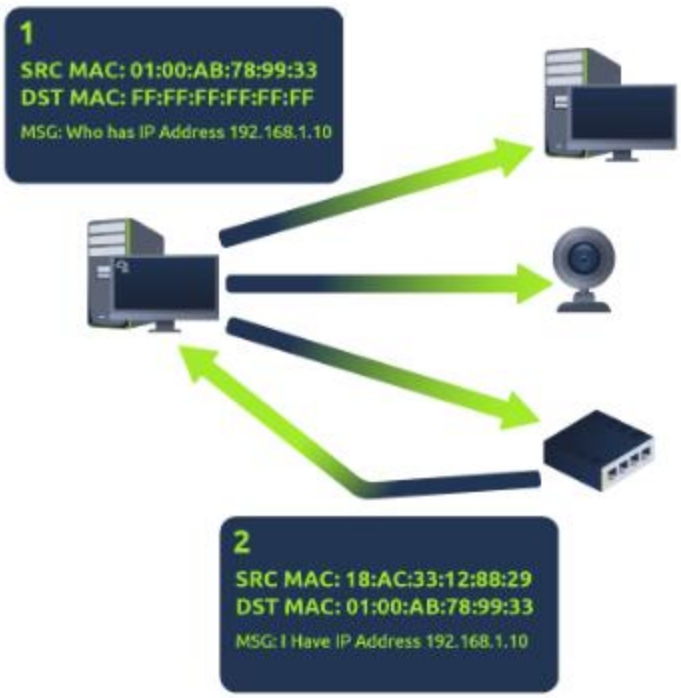
Each device within a network has a ledger to store information on, which is called a cache. In the context of the **ARP** protocol, this cache stores the identifiers of other devices on the network.

In order to map these two identifiers together (IP address and MAC address), the ARP protocol sends two types of messages:

1. **ARP Request**
2. **ARP Reply**

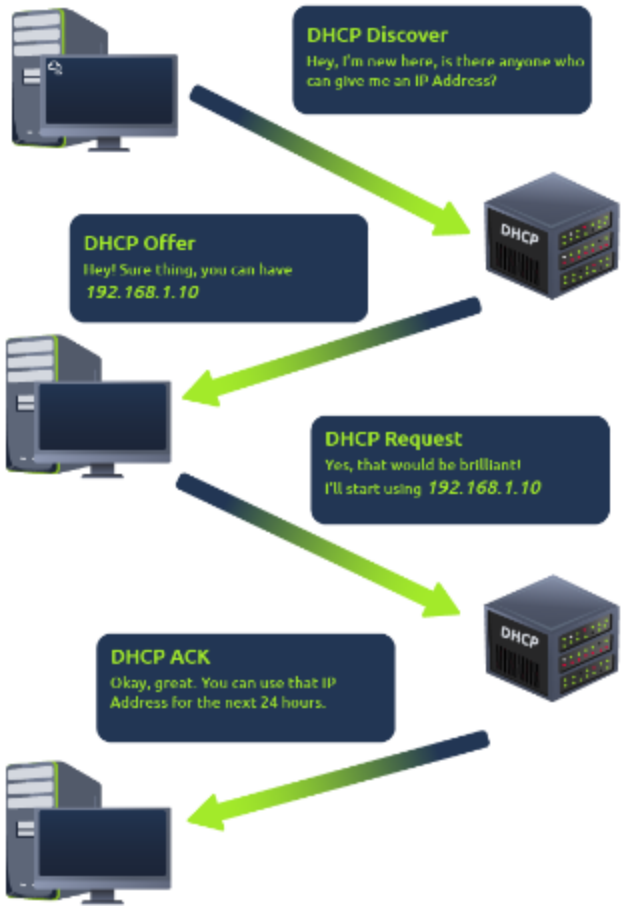
When an **ARP request** is sent, a message is broadcasted to every other device found on a network by the device, asking whether or not the device's MAC address matches the requested IP address. If the device does have the requested IP address, an **ARP reply** is returned to the initial device to acknowledge this. The initial device will now remember this and store it within its cache (an ARP entry).

This process is illustrated in the diagram below:



Task 4: The DHCP Protocol

IP addresses can be assigned either manually, by entering them physically into a device, or automatically and most commonly by using a **DHCP** (**D**ynamic **H**ost **C**onfiguration **P**rotocol) server. 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).



Room 3: OSI Model

Task 1: What is the OSI Model?

The **OSI** model (or **O**pen **S**ystems **I**nterconnection 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.

One of the main benefits of the OSI model is that devices can have different functions and designs on a network while communicating with other devices. Data sent across a network that follows the uniformity of the OSI model can be understood by other devices.

The OSI model consists of seven layers which are illustrated in the diagram below. Each layer has a different set of responsibilities and is arranged from Layer 7 to Layer 1.

At every individual layer that data travels through, specific processes take place, and pieces of information are added to this data, which is what we'll come to discuss in the upcoming tasks within this room. However, for now, we only need to understand that this process is called encapsulation.

Task 2 to Task 8: Layer 7– Application to Layer 1- Physical

Layer 7: Application: The application layer of the OSI model is the layer that you will be most familiar with. This familiarity is because 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.

Everyday applications such as email clients, browsers, or file server browsing software such as FileZilla provide a friendly, **G**raphical **U**ser **I**nterface (**GUI**) for users to interact with data sent or received. Other protocols include **DNS** (**D**omain **N**ame **S**ystem), which is how website addresses are translated into IP addresses.

Layer 6: Presentation: Layer 6 of the OSI model is the layer in which standardisation starts to take place. Because software developers can develop any software such as an email client differently, the data still needs to be handled in the same way — no matter how the software works.

This layer acts as a translator for data to and from the application layer (layer 7). The receiving computer will also understand data sent to a computer in one format destined for in another format. For example, when you send an email, the other user may have another email client to you, but the contents of the email will still need to display the same.

Security features such as data encryption (like HTTPS when visiting a secure site) occur at this layer.

Layer 5: Session: 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.

The session layer (layer 5) synchronises the two computers to ensure that they are on the same page before data is sent and received. Once these checks are in place, the session layer will begin to divide up the data sent into smaller chunks of data and begin to send these chunks (packets) one at a time. This dividing up is beneficial because if the connection is lost, only the chunks that weren't yet sent will have to be sent again — not the entire piece of the data (think of it as loading a save file in a video game).

What is worthy of noting is that sessions are unique — meaning that data cannot travel over different sessions, but in fact, only across each session instead.

Layer 4: Transport: Layer 4 of the OSI model plays a vital part in transmitting data across a network and can be a little bit difficult to grasp. When data is sent between devices, it follows one of two different protocols that are decided based upon several factors:

- TCP
- UDP

Let's begin with TCP. The **T**ransmission **C**ontrol **P**rotocol (**TCP**). Potentially hinted by the name, this protocol is designed with reliability and guarantee in mind. 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.

Not only this, but TCP incorporates error checking into its design. Error checking is how TCP can guarantee that data sent from the small chunks in the session layer (layer 5) has then been received and reassembled in the same order.

Let's summarise the advantages and disadvantages of TCP in the table below:

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 (no good having half a file!).

In the diagram below, we can see how a picture of a cat is broken down into small pieces of data (known as packets) from the "webserver", where the "computer" re-constructs the picture of the cat into the correct order.

Now let's move onto the **U**ser **D**atagram **P**rotocol (or **UDP** for short). This protocol is not nearly as advanced as its brother - the TCP protocol. It doesn't boast the many features offered by TCP, such as error checking and reliability. In fact, 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.

Whilst this sounds disadvantageous, it does have its merits, which we'll layout in the table below:

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.

Using the same example as before, we can now see that only Packets #1 and #3 have been received by the "Computer", meaning that half of the image is missing.

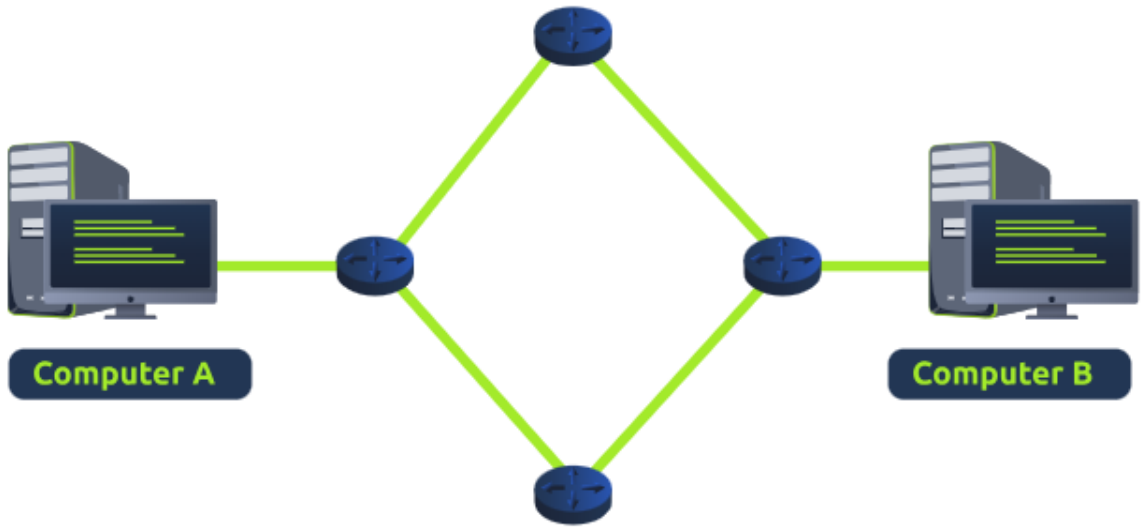
UDP is useful in situations where there are small pieces of data being sent. For example, protocols used for discovering devices (*ARP* and *DHCP)* or larger files such as video streaming (where it is okay if some part of the video is pixelated. Pixels are just lost pieces of data!)

Layer 3: Network: The third layer of the OSI model (network layer) is where the magic of routing & re-assembly of data takes place (from these small chunks to the larger chunk). Firstly, routing simply determines the most optimal path in which these chunks of data should be sent.

Whilst some protocols at this layer determine exactly what is the "optimal" path that data should take to reach a device, we should only know about their existence at this stage of the networking module. Briefly, these protocols include **OSPF** (**O**pen **S**hortest **P**ath **F**irst) and **RIP** (**R**outing **I**nformation **P**rotocol). 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.



Layer 2: Data Link: 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** (**M**edia **A**ccess **C**ontrol) address of the receiving endpoint. Inside every network-enabled computer is a **N**etwork **I**nterface Card (**NIC**) which comes with a unique MAC address to identify it.

MAC addresses are set by the manufacturer and literally burnt into the card; they can't be changed -- although they can be spoofed. When information is sent across a network, it's actually the physical address that is used to identify where exactly to send the information.

Additionally, it's also the job of the data link layer to present the data in a format suitable for transmission.

Layer 1: Physical: This layer is one of the easiest layers to grasp. Put simply, this layer references the physical components of the hardware used in networking and is the lowest layer that you will find. Devices use electrical signals to transfer data between each other in a binary numbering system (1's and 0's).

For example, ethernet cables connecting devices.

Room 4: Packets and Frames

Task 1: What are Packets and Frames?

Packets and frames are small pieces of data that, when forming together, make a larger piece of information or message. However, they are two different things in the OSI model. A frame is at layer 2 - the data link layer, meaning there is no such information as IP addresses. Think of this as putting an envelope within an envelope and sending it away. The first envelope will be the

packet that you mail, but once it is opened, the envelope within still exists and contains data (this is a frame).

This process is called encapsulation. At this stage, it's safe to assume that when we are talking about anything IP addresses, we are talking about packets. When the encapsulating information is stripped away, we're talking about the frame itself.

Packets are an efficient way of communicating data across networked devices such as those explained in Task 1. Because this data is exchanged in small pieces, there is less chance of bottlenecking occurring across a network than large messages being sent at once.

For example, when loading an image from a website, this image is not sent to your computer as a whole, but rather small pieces where it is reconstructed on your computer. Take the image below as an illustration of this process. The cat's picture is divided into three packets, where it is reconstructed when it reaches the computer to form the final image.

Packets have different structures that are dependant upon the type of packet that is being sent. As we'll come on to discuss, networking is full of standards and protocols that act as a set of rules for how the packet is handled on a device. With the Internet predicted to have approximately 50 billion devices connected by the end of 2020, things quickly get out of hand if there is no standardisation.

Let's continue with our example of the Internet Protocol. A packet using this protocol will have a set of headers that contain additional pieces of information to the data that is being sent across a network.

Some notable headers include:

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.

Task 2: TCP/IP (The Three -Way Handshake)

TCP (or **T**ransmission **C**ontrol **P**rotocol for short) is another one of these rules used in networking.

This protocol is very similar to the OSI model that we have previously discussed in room three of this module so far. The TCP/IP protocol consists of four layers and is arguably just a summarised version of the OSI model. These layers are:

- Application
- Transport
- Internet
- Network Interface

Very similar to how the OSI model works, information is added to each layer of the TCP model as the piece of data (or packet) traverses it. As you may recall, this process is known as encapsulation - where the reverse of this process is decapsulation.

One defining feature of TCP is that it is **connection-based**, which means that TCP must establish a connection between both a client and a device acting as a server **before** data is sent.

Because of this, TCP guarantees that any data sent will be received on the other end. This process is named the Three-way handshake, which is something we'll come on to discuss shortly. A table comparing the advantages and disadvantages of TCP is located below:

Advantages of TCP	Disadvantages of TCP
Guarantees the integrity 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 and must be re-sent.
Capable of synchronising two devices to prevent each other from being flooded with data in the wrong order.	A slow connection can bottleneck another device as the connection will be reserved on the other device the whole time.
Performs a lot more processes for reliability	TCP is significantly slower than UDP because more work (computing) has to be done by the devices using this protocol.

TCP packets contain various sections of information known as headers that are added from encapsulation. Let's explain some of the crucial headers in the table below:

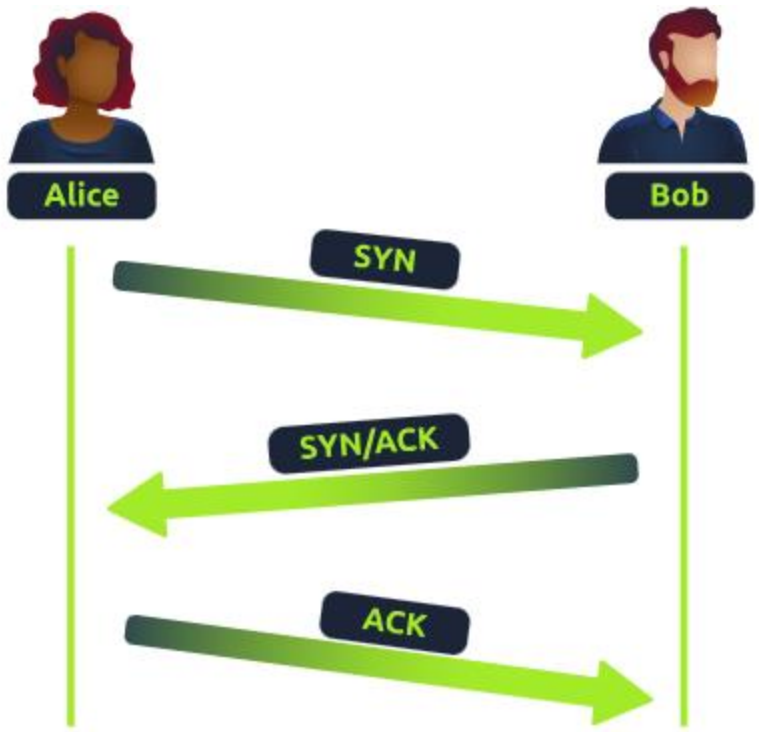
Header	Description

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.

Next, we'll come on to discuss the *Three-way handshake* - the term given for the process used to establish a connection between two devices. The Three-way handshake communicates using a few special messages - the table below highlights the main ones:

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.

The diagram below shows a normal Three-way handshake process between Alice and Bob. In real life, this would be between two devices.



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 **SYN**chronise with (0)
2. SYN/ACK - Server: Here's my Initial Sequence Number (ISN) to **SYN**chronise with (5,000), and I **ACK**nowledge your initial number sequence (0)

3. ACK - Client: I **ACK**nowledge 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

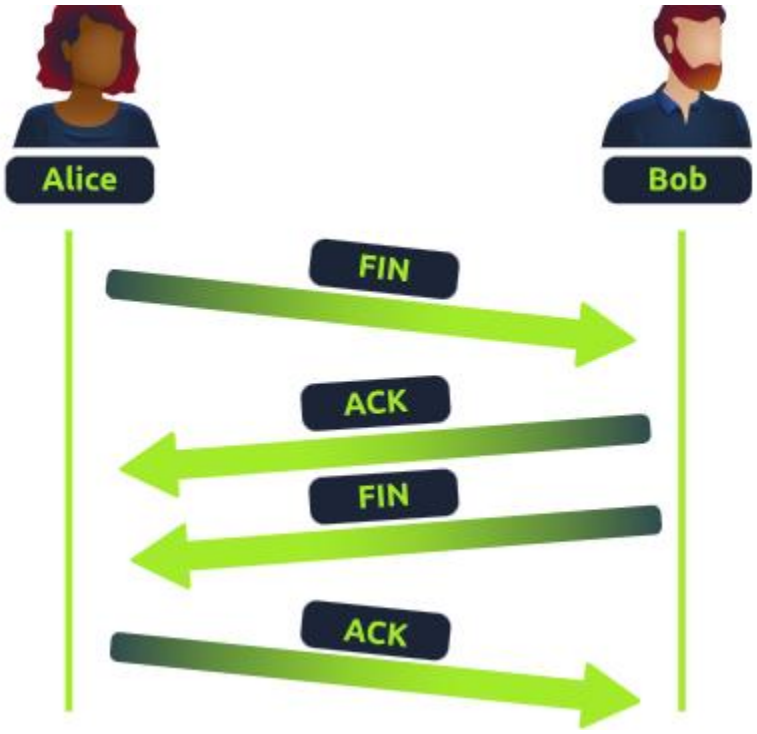
TCP Closing a Connection:

Let's quickly explain the process behind TCP closing a connection. First, TCP will close a connection once a device has determined that the other device has successfully received all of the data.

Because TCP reserves system resources on a device, it is best practice to close TCP connections as soon as possible.

To initiate the closure of a TCP connection, the device will send a "FIN" packet to the other device. Of course, with TCP, the other device will also have to acknowledge this packet.

Let's show this process using Alice and Bob as we have previously.



In the illustration, we can see that Alice has sent Bob a "**FIN**" packet. Because Bob received this, he will let Alice know that he received it and that he also wants to close the connection (using FIN). Alice has heard Bob loud and clear and will let Bob know that she acknowledges this.

Task 4: UDP/IP

The **U**ser **D**atagram **P**rotocol (**UDP**) is another protocol that is used to communicate data between devices.

Unlike its brother TCP, UDP is a **stateless** protocol that doesn't require a constant connection between the two devices for data to be sent. For example, the Three-way handshake does not occur, nor is there any synchronisation between the two devices.

Recall some of the comparisons made about these two protocols in Room 3: "OSI Model". Namely, UDP is used in situations where applications can tolerate data being lost (such as video streaming or voice chat) or in scenarios where an unstable connection is not the end-all. A table comparing the advantages and disadvantages of UDP is located below:

Advantages of UDP	Disadvantages of UDP
UDP is much faster than TCP.	UDP doesn't care if the data is received or not.
UDP leaves the application (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.

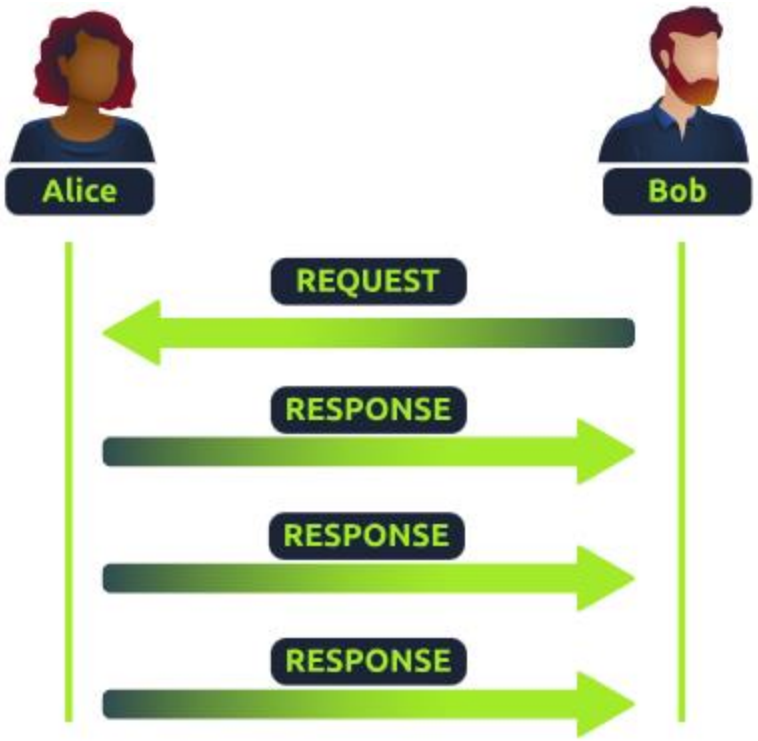
As mentioned, no process takes place in setting up a connection between two devices. Meaning that there is no regard for whether or not data is received, and there are no safeguards such as those offered by TCP, such as data integrity.

UDP packets are much simpler than TCP packets and have fewer headers. However, both protocols share some standard headers, which are what is annotated in the table below:

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 UDP 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.

Next, we'll come on to discuss how the process of a connection via UDP differs from that of something such as TCP. We should recall that UDP is **stateless**. No acknowledgement is sent during a connection.

The diagram below shows a normal UDP connection between Alice and Bob. In real life, this would be between two devices.



Task 5: Ports

Perhaps aptly titled by their name, ports are an essential point in which data can be exchanged. Think of a harbour and port. Ships wishing to dock at the harbour will have to go to a port compatible with the dimensions and the facilities located on the ship. When the ship lines up, it will connect to a **port** at the harbour. Take, for instance, that a cruise liner cannot dock at a port made for a fishing vessel and vice versa.

These ports enforce what can park and where — if it isn't compatible, it cannot park here. Networking devices also use ports to enforce strict rules when communicating with one another. When a connection has been established (recalling from the OSI model's room), any data sent or received by a device will be sent through these ports. In computing, ports are a numerical value between **0** and **65535** (65,535).

Because ports can range from anywhere between 0-65535, there quickly runs the risk of losing track of what application is using what port. A busy harbour is chaos! Thankfully, we associate applications, software and behaviours with a standard set of rules. For example, by enforcing that any web browser data is sent over port 80, software developers can design a web browser such as Google Chrome or Firefox to interpret the data the same way as one another.

This means that all web browsers now share one common rule: data is sent over port 80. How the browsers look, feel and easy to use is up to the designer or the user's decision.

While the standard rule for web data is *port 80*, a few other protocols have been allocated a standard rule. Any port that is within **0** and **1024** (1,024) is known as a common port. Let's explore some of these other protocols below:

Protocol	Port Number	Description
File **T**ransfer **P**rotocol (**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 **Sh**ell (**SSH**)	22	This protocol is used to securely login to systems via a text-based interface for management.
Hyper**T**ext 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.
Hyper**T**ext **T**ransfer **P**rotocol **S**ecure (**HTTPS**)	443	This protocol does the exact same as above; however, securely using encryption.
Server **M**essage **B**lock (**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 **D**esktop **P**rotocol (**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).

What is worth noting here is that these protocols only follow the standards. i.e. you can administer applications that interact with these protocols on a different port other than what is the standard (running a web server on 8080 instead of the 80 standard port). Note, however, applications will presume that the standard is being followed, so you will have to provide a **colon (:)** along with the port number.