What is email infrastructure?

Email infrastructure is the set of software and hardware components that are triggered as soon as you write an email and hit the send button. It combines mail servers, agents, and IP addresses – basically everything that you'd need for the successful delivery of email campaigns.

Think of it as a postal system that combines postal offices, staff who sort through mail, in-house or third-party delivery services, and postal carriers who bring mail to your doorstep.

Even though emails are written and sent through the internet, the structure of the infrastructure is rather similar to real-life postal services, a physical location where you can store your servers and IT assets (excluding all servers and authentication protocols, of course). And just as you would implement security measures to protect your postal service office from any incident, you would also need to protect your servers and assets in a sustainable data center to ensure their uptime.

Email infrastructure architecture

Email infrastructure consists of the following elements: mail agents (MSA, MTA, MDA, MUA), email servers (send mail and receive them – SMTP server responsible for sending and delivering mails. IMAP and POP3 used to retrieve mails from receiving server), authentication protocols (DKIM, SOF, DMARC, BIMI), IP addresses (vital components of email infrastaructure, that is determiend by DNS. An Ip can be "shared IP" or "dedicated IP". But each of them significantly impacts the sender reputatuion and email deliverability) , sending domains (tells the server wo is sending email – varming up), and feedback loops (3th party analytics, scaling) .

Mail Agends

Mail Agends are component of email infrastructure (Email infrastructure consists of the following elements: mail agents, mail servers, authentication protocols, IP addresses, sending domains, and feedback loops.) that handle the whole process of sending and delivering emails.

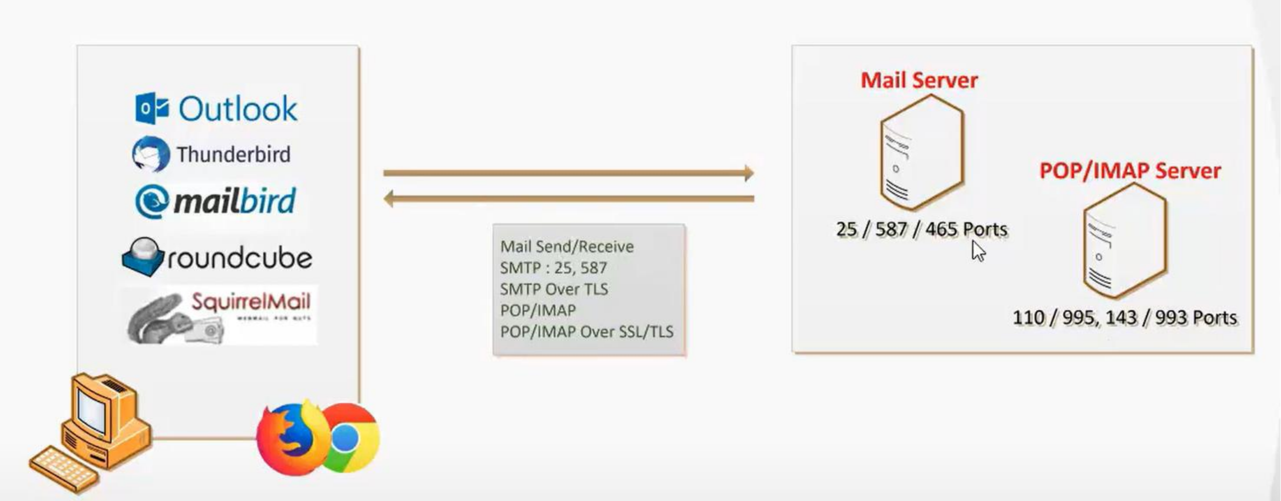
There are four types of mail agents are used to send, deliver, and retrieve emails. These are; Mail User Agents, Mail Submission Agents, Mail Transfer Agents and Mail Delivery Agents.

MUA
MTA
MUA
MSA
MDA

A Mail User Agent (email client)

A Mail User Agent (MUA) is a software application that enables users to send, receive, and manage their email messages. It acts as an interface between the user and the mail server, facilitating communication through protocols like SMTP for sending emails and IMAP or POP3 for receiving them. The MUA often includes features like composing messages, organizing mail folders, and managing contacts, enhancing the overall user experience in electronic communication.

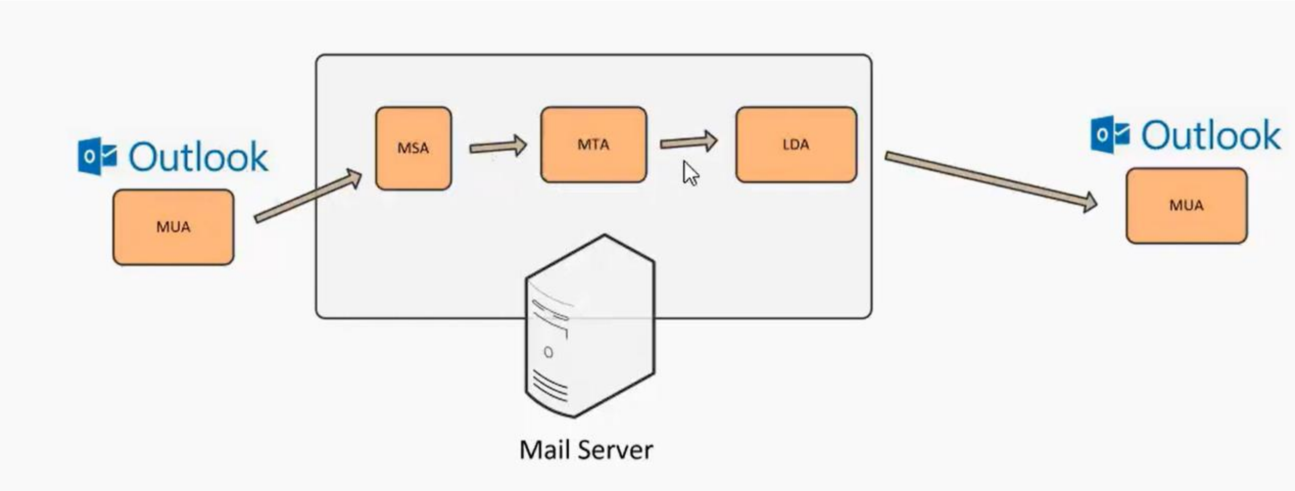
- Mail User Agents can be web-based like Gmail or desktop applications like Microsoft Outlook.
- MUAs handle email formatting and attachment management, ensuring that messages are properly displayed across different platforms.
- They often support multiple email accounts, allowing users to consolidate their communication in one place.
- Many MUAs incorporate spam filtering features to help manage unwanted emails.
- The user interface of an MUA is designed to be intuitive, providing easy navigation through inboxes, sent items, and folders.



Mail Submission Agent (MSA)

This is a program inside the mail server package which listens on port 587 and it is for receiving mail from MUA. MSA talks with ESMTP extended SMTP with a mail client and it doesn't allow sending mail without authentication that's why it is more secure.

Many ISP blocks 25 port to connect from MUA for spam prediction. In that case we have to connect MSA for email delivery. MSA recieves mail from MUA and deliver to MTA for final delivery. It is not mandotory to use MSA 587 port but highly recommended. SMTP 25 port is for server to server and MSA 587 port is MUA to MTA commincation.



Mail Transport Agent (MTA)

MTA is the backbone of an email system. It takes emails from UA or MSA, understands the recipient's address, and delivers them to their destination. MTA uses Simple Mail Transport Protocol (SMTP) for its operation.

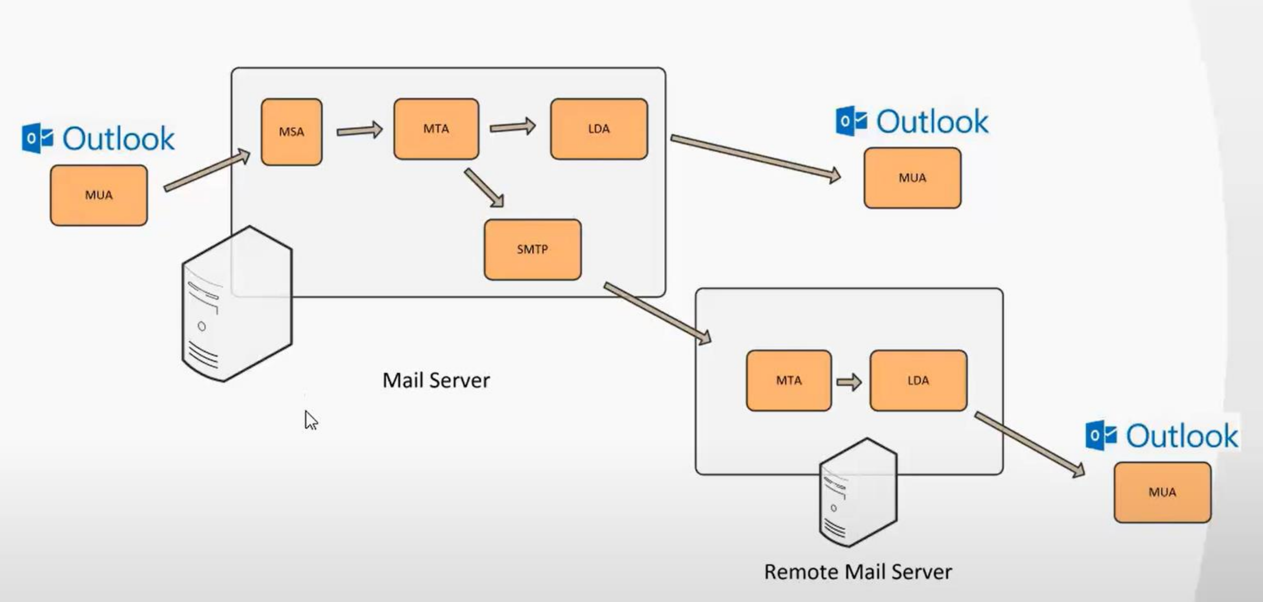
Essential MTA operations are listed below:

- Outgoing mail operation
- Receive outgoing emails from UA and MSA
- Understand the recipient's addresses
- If require, rewrite the addresses
- Perform a DNS MX lookup to resolve the recipient's address
- If the recipient is directly accessible, forward the email to MDA
- If the recipient is not directly accessible, forward the email to the next responsible MTA

Incoming mail operation

- Receive incoming mails from other MTA
- Understand the recipient's addresses
- If the recipient belongs to it, perform a virus scan and forward the email to MDA
- If the recipient does not belong to it, forward the email to the next responsible MTA

Basic Example



MTA receives mail from MUA or MSA and delivered to LDA (Local Delivery Agent) and remote mail to SMPT (remote delivery agent). LDA stores the mail in local mail user box, SMPT communicate with remote mail server (MTA) to deliver the mail. MTA is not single delivery agent, it is the main engine.

What MTA-STS?

Mail Transfer Agent Strict Transport Security (MTA-STS) is an email security standard for secure delivery of email to your domain. With MTA-STS you let senders know that the inbound email service for your domain accepts secure email delivery using SMTP over TLS (STARTTLS), and that email should not be delivered over an insecure SMTP connection.

MTA-STS mitigates Man-In-The-Middle DNS and SMTP downgrade attacks that would allow an attacker to read or manipulate email in transit (Basic explanation What is MTA-STS ? (2024)) .

Why MTA-STS?

When an email is sent, the sending email service (MTA) will determine where to deliver the email to by querying the DNS for the MX records of the receiving domain. For example: an MTA which must deliver an email to example@mailhardener.com will perform a DNS query for the MX records of mailhardener.com to learn which MTA the email must be sent to. The MTA then connects to the MTA it found in the DNS query result and negotiates if this MTA supports the STARTTLS command. If it does, the MTAs switch to an encrypted connection, and the email is then delivered securely.

If the receiver does not support STARTTLS, or the secure connection cannot be established for other reasons, the email is sent using an unencrypted connection.

There are two Man-In-The-Middle (MITM) attacks possible here:

The attacker could inject a malicious DNS response for the MX query, tricking the sending MTA to deliver email to an MTA controlled by the attacker. This is known as a DNS spoofing attack.

The STARTTLS negotiation could be disrupted, tricking the sending MTA into sending the email without the use of TLS encryption. This is known as a downgrade attack.

Both attacks result in the attacker being able to read and manipulate the email while it is being sent.

MTA-STS aims to mitigate both vulnerabilities, it is specified in RFC8461.

DNS spoofing attacks

DNS is an unencrypted protocol, for anyone who operates a portion of the internet (like ISPs, governments, public Wi-Fi, etc.) it is trivial to manipulate DNS traffic that flows through the network. By doing so, an attacker can replace the MX records in a DNS query response with an email service that they control. This results in the sending MTA delivering the email straight to the attacker controlled email host. Once the attacker has copied or even manipulated the email, it is then forwarded to the real MX for that domain, as if nothing happened.

Because MX records of a domain point to MTAs by name, not IP address, the attacker can use a domain that they control. This enables the attacker to provide a valid certificate, so STARTTLS will not protect against DNS based attacks.

An attack where DNS responses are manipulated is known as a DNS spoofing attack. Attempts have been made to protect against DNS spoofing by introducing zone signing in the DNSSEC (RFC4033) standard. But concerns have been raised about the security and practicality of DNSSEC to a point where some even advise against using DNSSEC.

With MTA-STS, the MX addresses fetched over DNS are compared to those found in the MTA-STS policy file, which is served over a secure HTTPS connection. And because the MTA-STS policies are also cached by MTAs, a successful DNS spoofing attack becomes very unlikely. *(Really good webinar Zimbra Email Security Webinar Series: MTA STS, TLS RPT, and BIMI - YouTube).*

Message Delivery Agent (MDA)

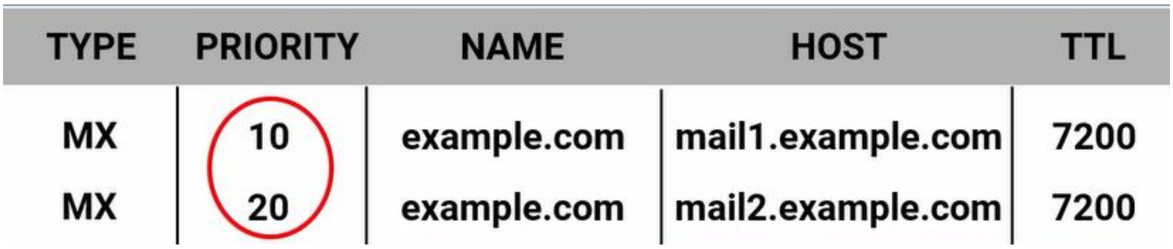
Once the message goes through all the MTAs, it is passed to the message delivery agent, which converts received messages into the appropriate format and transfers them to the recipient’s MUA. In the best-case scenario, messages will end up in the recipient’s inbox. However, if the sender’s domain reputation is low or the authentication fails, messages will go to the spam folder or get discarded.

MDA would be the mail carrier who picks up mail from the recipient’s post office and delivers it to their mailbox (MUA).

MX record

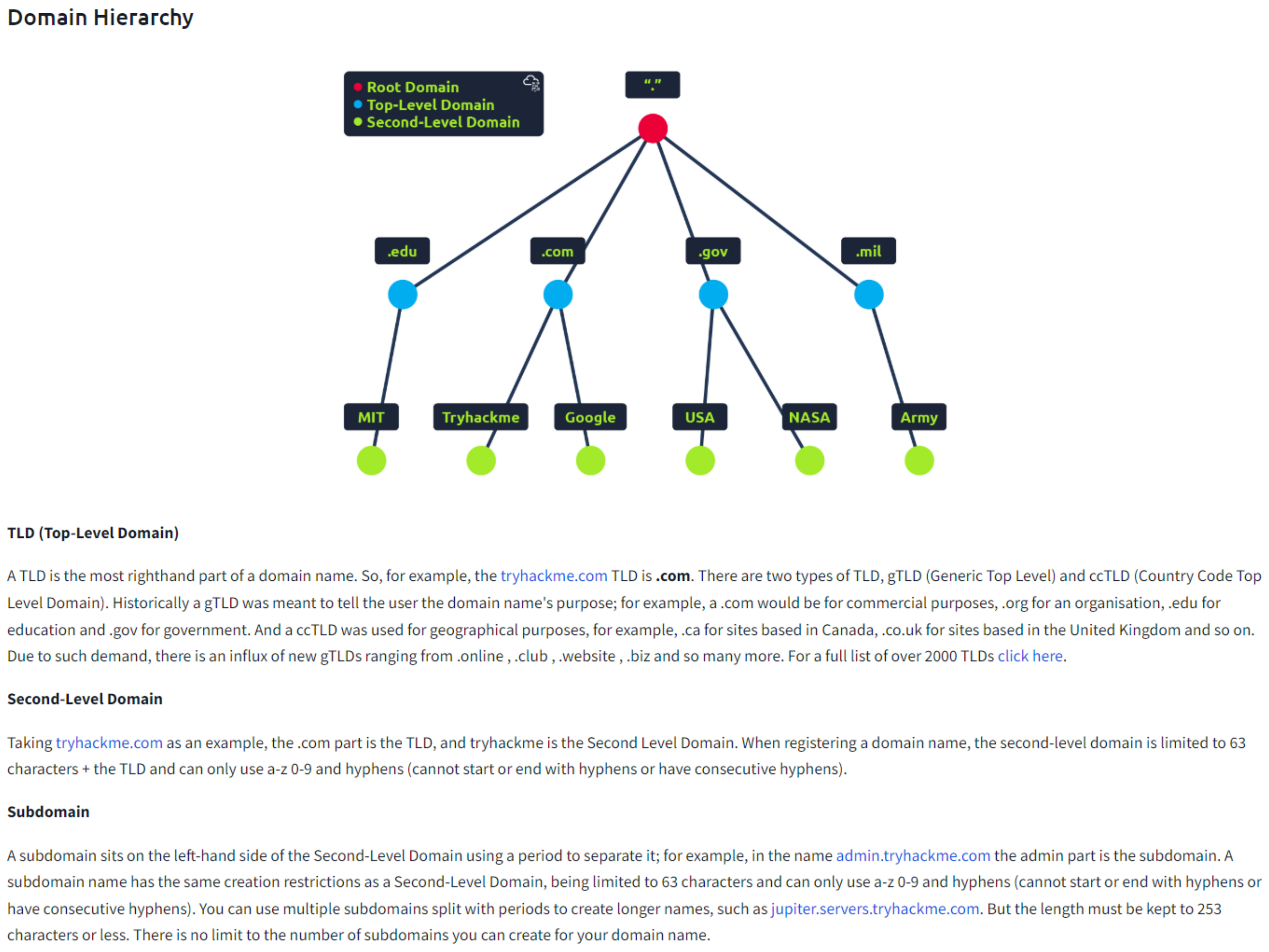
Mail Exchanger (MX) records are used to create email addresses from that domain. The MX record simply points to the server where emails should be delivered for that domain name. When you send email to tom@example.com, your MTA (Mail Transfer Agent) will quey the MX records for example.com because it's looking for an email server and DNS will respond will back telling MTA which server to send the email to which in this case would mail1.example.com because that's what the MX records points to. MX record tells the world which server to send email to for a particular domain name.

MX records generally have two entries, a primary email server and secondary email server along with priority numbers. The lower the priority number means that it's the primary server but if the primary email server gets overwhelmed or goes down, then the secondary email server would be used



TYPE	PRIORITY	NAME	HOST	TTL
MX	10	example.com	mail1.example.com	7200
MX	20	example.com	mail2.example.com	7200

Just a refresher



Domain Hierarchy

TLD (Top-Level Domain)

A TLD is the most righthand part of a domain name. So, for example, the tryhackme.com TLD is **.com**. There are two types of TLD, gTLD (Generic Top Level) and ccTLD (Country Code Top Level Domain). Historically a gTLD was meant to tell the user the domain name's purpose; for example, a .com would be for commercial purposes, .org for an organisation, .edu for education and .gov for government. And a 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. Due to such demand, there is an influx of new gTLDs ranging from .online , .club , .website , .biz and so many more. For a full list of over 2000 TLDs click here.

Second-Level Domain

Taking tryhackme.com as an example, the .com part is the TLD, and tryhackme is the Second Level Domain. 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.

DNS Record Types

DNS isn't just for websites though, and multiple types of DNS record exist. We'll go over some of the most common ones that you're likely to come across.

A Record

These records resolve to IPv4 addresses, for example 104.26.10.229

AAAA Record

These records resolve to IPv6 addresses, for example 2606:4700:20::681a:be5

CNAME Record

These records resolve to another domain name, for example, TryHackMe's online shop has the subdomain name store.tryhackme.com which returns a CNAME record shops.shopify.com. Another DNS request would then be made to shops.shopify.com to work out the IP address.

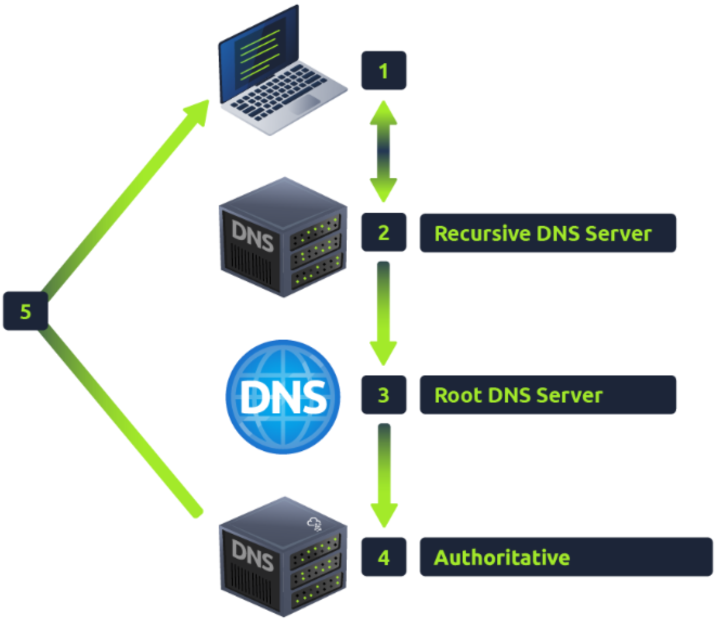
MX Record

These records resolve to the address of the servers that handle the email for the domain you are querying, for example an MX record response for tryhackme.com would look something like alt1.aspmx.l.google.com. These records also come with a priority flag. This tells the client in which order to try the servers, this is perfect for if the main server goes down and email needs to be sent to a backup server.

TXT Record

TXT records are free text fields where any text-based data can be stored. TXT records have multiple uses, but some common ones can be to list servers that have the authority to send an email on behalf of the domain (this can help in the battle against spam and spoofed email). They can also be used to verify ownership of the domain name when signing up for third party services.

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.