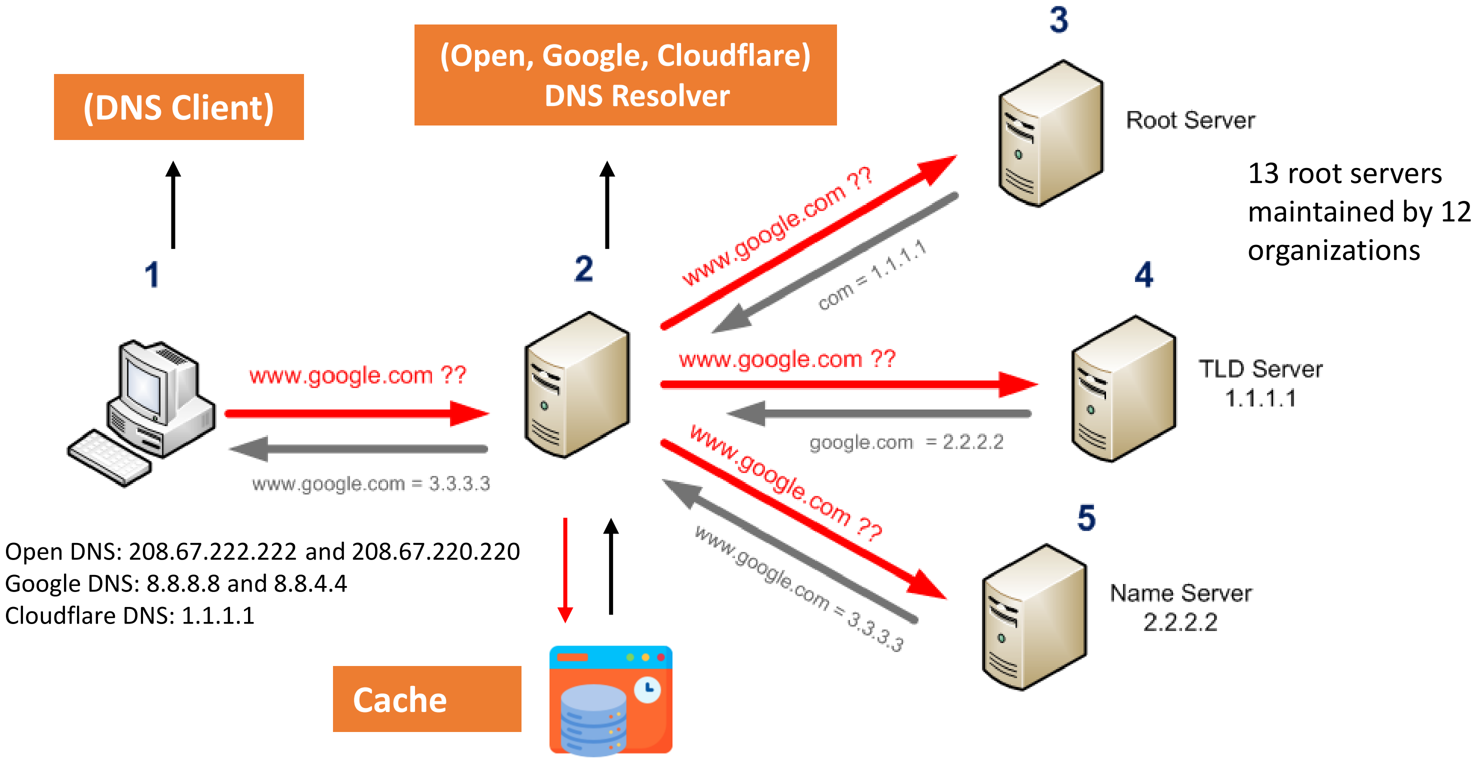
DNS (Domain Name System)

→ DNS is an application hosted on a computer server which holds the database of the internet protocol addresses mapped to their domain/host names and it is used to translate domain names to IP address and vice versa.

→ It works on application layer of the OSI model.

→ It uses TCP port no 53 or UDP port no 53.



(DNS Client)

(Open, Google, Cloudflare) DNS Resolver

Cache

13 root servers maintained by 12 organizations

Open DNS: 208.67.222.222 and 208.67.220.220
Google DNS: 8.8.8.8 and 8.8.4.4
Cloudflare DNS: 1.1.1.1

DNS Client – The browser that initiates the DNS query to find the IP address of unknown domain names.

DNS Resolver – A server (not a device) that resolves domain names into IP addresses. It performs non-recursive or iterative queries on behalf of the client to locate the IP address of the requested domain name.

(

non recursive – single query and reply - resolver to root
non recursive – single query and reply - resolver to TLD
non recursive – single query and reply - resolver to Authoritative } Iterative

)

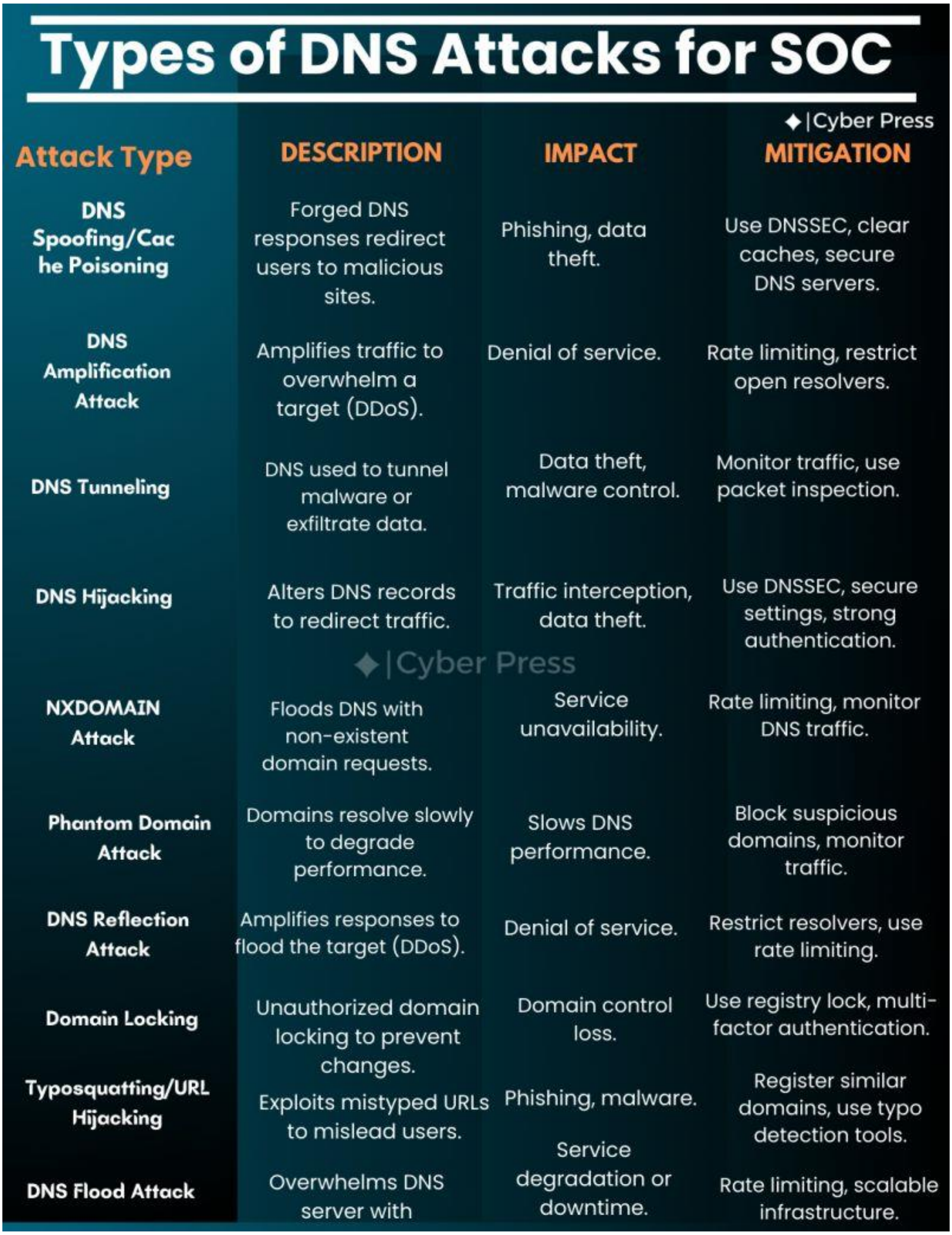
https://www.cloudflare.com/en-in/learning/dns/dns-security/

DNS Resolution Process

1. You type a domain (e.g., cloudflare.com) in the browser.
2. Browser checks its browser cache
3. If an IP address is found → request is sent directly to the web server.
   If not found in the browser cache, the browser sends the query to the OS resolver.
4. The OS resolver follows this order:
   i. First, checks the local Hosts file (e.g.,C:\Windows\System32\drivers\etc\hosts on Windows or /etc/hosts on Linux).
   ii. Then, checks the OS-level DNS cache.
   iii.If still not found, it forwards the DNS query to the configured external DNS resolver (like 1.1.1.1)
5. External resolver performs full DNS resolution (recursive process).
   i. At first, the resolver sends the query to the Root server to get the IP address of the FQDN.
   (resolver sends query to get the IP address of the cloudflare.com)
   ii. Root server replies with IP address of the TLD server
   (it replies with the Ip address of the **.com TLD** server)
   iii. Resolver sends the query to the TLD server to get the IP address of the FQDN.
   iv. TLD Server replies with the Ip address of the authoritative name server.
   v. The resolver sends the query to the Authoritative Name server to get the IP address of the FQDN.
   vi. Authoritative name Server replies with the Ip address of FQDN.
6. The IP is obtained and stored in the DNS cache, it is returned back to the browser
7. The browser performs TCP 3way handshake to check integrity of the server. To check whether DNS server is the one, it wants to connect with.
8. Last but not the least, The browser performs TLS handshake with the server. So that they will verify each other, and change their data into encrypted format.
9. Finally, they are going to exchange data.

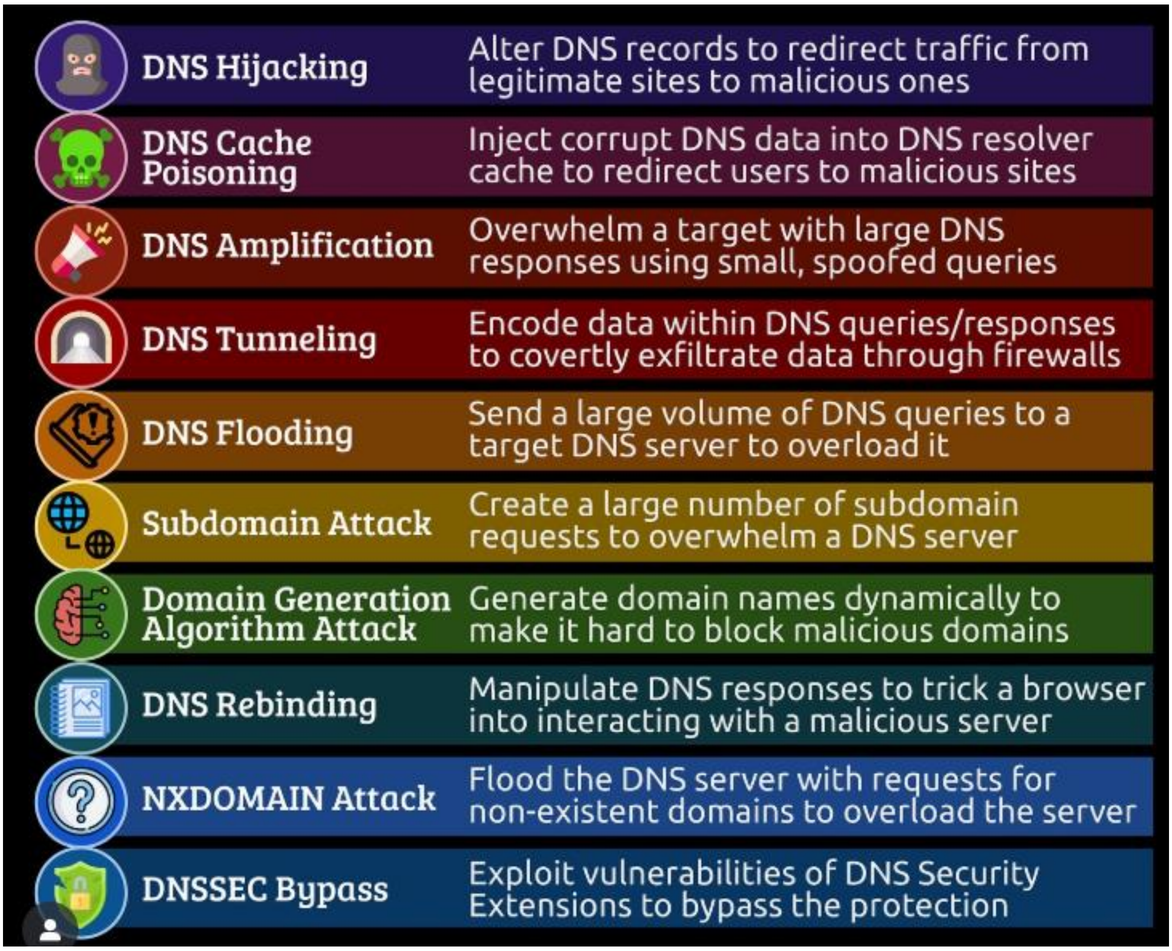
Why DNS security is important? And Types of DNS attack

ANS: DNS plays an important role while a device is accessing the internet. The DNS server communicates with a unique and unsigned packet, that makes it is vulnerable for attacks.



Types of DNS Attacks for SOC

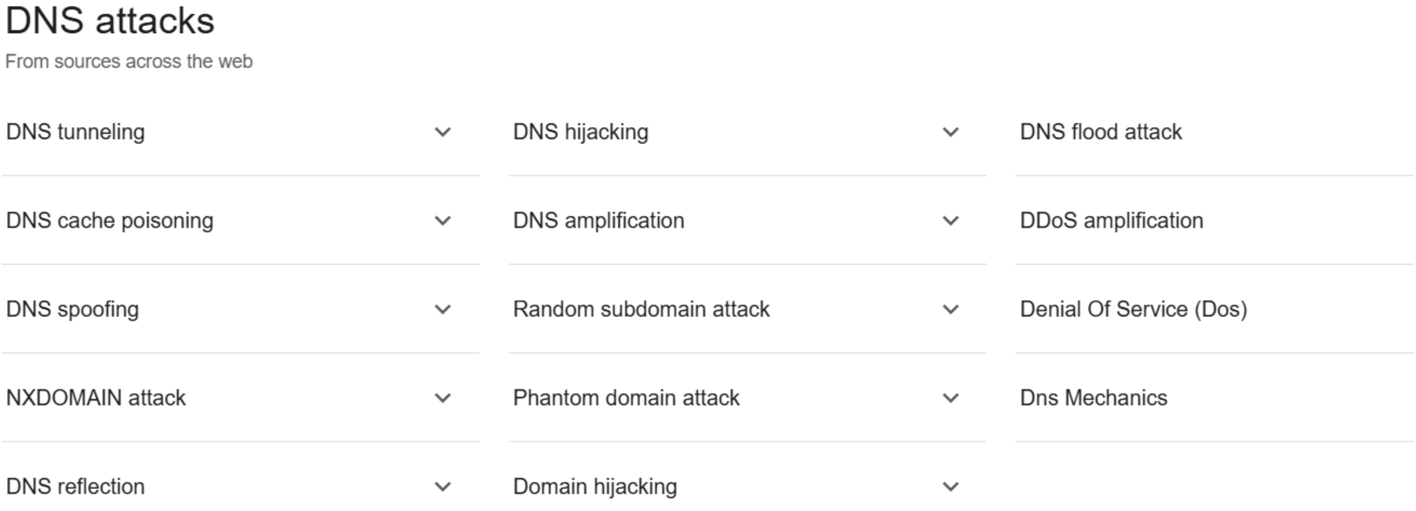
Attack Type	DESCRIPTION	IMPACT	MITIGATION
DNS Spoofing/Cache Poisoning	Forged DNS responses redirect users to malicious sites.	Phishing, data theft.	Use DNSSEC, clear caches, secure DNS servers.
DNS Amplification Attack	Amplifies traffic to overwhelm a target (DDoS).	Denial of service.	Rate limiting, restrict open resolvers.
DNS Tunneling	DNS used to tunnel malware or exfiltrate data.	Data theft, malware control.	Monitor traffic, use packet inspection.
DNS Hijacking	Alters DNS records to redirect traffic.	Traffic interception, data theft.	Use DNSSEC, secure settings, strong authentication.
NXDOMAIN Attack	Floods DNS with non-existent domain requests.	Service unavailability.	Rate limiting, monitor DNS traffic.
Phantom Domain Attack	Domains resolve slowly to degrade performance.	Slows DNS performance.	Block suspicious domains, monitor traffic.
DNS Reflection Attack	Amplifies responses to flood the target (DDoS).	Denial of service.	Restrict resolvers, use rate limiting.
Domain Locking	Unauthorized domain locking to prevent changes.	Domain control loss.	Use registry lock, multi-factor authentication.
Typosquatting/URL Hijacking	Exploits mistyped URLs to mislead users.	Phishing, malware.	Register similar domains, use typo detection tools.
DNS Flood Attack	Overwhelms DNS server with	Service degradation or downtime.	Rate limiting, scalable infrastructure.



DNS Hijacking	Alter DNS records to redirect traffic from legitimate sites to malicious ones
DNS Cache Poisoning	Inject corrupt DNS data into DNS resolver cache to redirect users to malicious sites
DNS Amplification	Overwhelm a target with large DNS responses using small, spoofed queries
DNS Tunneling	Encode data within DNS queries/responses to covertly exfiltrate data through firewalls
DNS Flooding	Send a large volume of DNS queries to a target DNS server to overload it
Subdomain Attack	Create a large number of subdomain requests to overwhelm a DNS server
Domain Generation Algorithm Attack	Generate domain names dynamically to make it hard to block malicious domains
DNS Rebinding	Manipulate DNS responses to trick a browser into interacting with a malicious server
NXDOMAIN Attack	Flood the DNS server with requests for non-existent domains to overload the server
DNSSEC Bypass	Exploit vulnerabilities of DNS Security Extensions to bypass the protection

How to Protect Against DNS Attacks

✅ **Use DNSSEC** (prevents spoofing & cache poisoning).
✅ **Enable DNS filtering** (blocks malicious domains).
✅ **Use encrypted DNS** (DoH/DoT – DNS over HTTPS/TLS).
✅ **Monitor DNS traffic** for anomalies.
✅ **Patch DNS servers & routers** regularly.
✅ **Use rate limiting** to prevent DDoS.



DNS attacks

From sources across the web

DNS tunneling
DNS cache poisoning
DNS spoofing
NXDOMAIN attack
DNS reflection

DNS hijacking
DNS amplification
Random subdomain attack
Phantom domain attack
Domain hijacking

DNS flood attack
DDoS amplification
Denial Of Service (Dos)
Dns Mechanics

(1) Man-In-The-Middle Attack (client-resolver)

→In a Man-in-the-Middle (MitM) attack, the attacker responds to a client's DNS query by spoofing the IP address of the DNS resolver.

→This type of attack is possible on a Local Area Network (LAN), especially when the attacker has access to a compromised or unsecured Wi-Fi network.

→This attack is realistic only if the attacker is in a position to sniff traffic, such as being connected to the same Wi-Fi network as the victim.

→This allows the attacker to redirect the client to a malicious IP address, potentially leading to phishing, malware installation, or data theft.

→There are two types of the man-in-the-middle attack: active or passive. In active an actor can modify the DNS data, in passive mode actor can only views the unsecured traffic.

(2) Cache poisoning or Spoofing (Resolver)

An attacker inserts a false DNS record into the resolver's cache. When a user queries the same domain, the resolver returns the fake IP address. As a result, the user is redirected to a malicious website controlled by the attacker, where sensitive information can be stolen.
Impact: Phishing, malware distribution

(3) Tunneling (Inside organization)

In DNS tunneling, an attacker inside the organization can exfiltrate sensitive data by encoding it into DNS queries that are sent to an external DNS server controlled by the attacker.
Can be blocked using deep packet inspection on the firewall
Impact: Data theft, malware communication.

(4) DNS Amplification (host device/network)

→ An attacker can send a large number of DNS replies to a host system, overwhelming it and causing a Denial-of-Service (DoS).
Prevent: Network-level DoS protection (e.g., IPS/IDS)

→ An attacker sends the request by spoofing the IP of targeted device, and the DNS server sends large amount of data to the host.

Impact: Website/service downtime.

(5) Hijacking (on name server)

In this scenario, an attacker reroutes DNS queries to a different domain name server. This can be achieved with malware infection or unauthorized alteration of a DNS server. While the outcome is similar to that of DNS spoofing, the method differs as DNS hijacking specifically targets the DNS record of the website on its nameserver.
Local DNS hijack – Changing the resolver address on host device
Router DNS hijacking – compromised public WIFI or Router

(6) Flood Attack

In this type of attack, an attacker sends a huge amounts of requests to the DNS server, overwhelm it, and making it unresponsive.
Impact: DNS server crashes, disrupting internet access.

(7) Phantom Domain Attack

What it does: A Phantom DNS Attack is a type of Denial-of-Service (DoS) attack that targets recursive DNS resolvers by forcing them to connect to malicious or unresponsive authoritative DNS servers, which reply very slowly or not at all.

(8) NXDOMAIN Attack

What it does: Overload the authoritative DNS server by making it respond to millions of **non-existent domain** queries.
Impact: Degrades DNS performance, causes timeouts.

(9) Random Subdomain Attack (t: nx att)

→ A **specific type** of NXDOMAIN attack.
→ Attacker sends queries for **non-existent subdomains** of a real domain: These subdomains **don’t exist**, so the **authoritative server** must check each one — and return NXDOMAIN

(10) Domain Lock-Up attack

It exhausts the resources of a DNS resolver (like memory, sockets, or threads) by forcing it to connect to malicious authoritative DNS servers that never reply — or reply very slowly.

The attacker controls a malicious authoritative DNS server, or registers a domain (e.g., malicious.com) and points it to their own DNS server.

A Domain Lock-Up Attack forces a DNS resolver to waste resources by querying intentionally slow or silent authoritative servers, eventually causing it to lock up or crash.

(11) Botnet

A Botnet is a network of infected devices (called bots or zombies) that are secretly controlled by an attacker (called a botmaster) — usually without the owner's knowledge.

These devices can include:

→ Home routers
→ CPEs (modems, set-top boxes)
→ IoT devices (smart cameras, thermostats)
→ Infected PCs, servers, or mobile phones

The goal is to exhaust the authoritative DNS server and cause a Denial-of-Service for the target website.

(DNS SECURITY)

→ DNSSEC (DNS Security Extensions) is a suite of security protocols designed to protect the Domain Name System (DNS) from attacks like cache poisoning, spoofing, and man-in-the-middle (MITM) attacks. It does this by adding cryptographic signatures to DNS records, ensuring data integrity and authenticity.

→DNSSEC uses **public-key cryptography** to sign DNS records, creating a chain of trust from the root DNS servers down to the final domain.

What are other ways of protecting against DNS-based attacks?

i. **Load balancing** the traffic across multiple DNS servers when one begins to perform poorly.

ii. **DNS Firewall:**

It is a tool placed in between DNS resolver and nameserver of the website or service they are trying to reach. The firewall can provide rate limiting services to shut down attackers trying to overwhelm the server. If the server does experience downtime as the result of an attack or for any other reason, the DNS firewall can keep the operator's site or service up by serving DNS responses from cache.

iii.**DNS OVER TLS & HTTPS:**

the most important difference between DoT and DoH is what port they use. DoT only uses port 853, while DoH uses port 443, which is the port that all other HTTPS traffic uses as well.

NOTE:

DNSSEC is a set of security extensions for verifying the identity of DNS Root Servers and authoritative nameservers in communications with DNS Resolvers. It is designed to prevent DNS cache poisoning, among other attacks. It does not encrypt communications. DNS over TLS or HTTPS, on the other hand, does encrypt DNS queries. 1.1.1.1 supports DNSSEC as well.