Name: Erfan Ahmadbeigi

Master :Mr.Milad nourian nasab

Subject:DNS attack

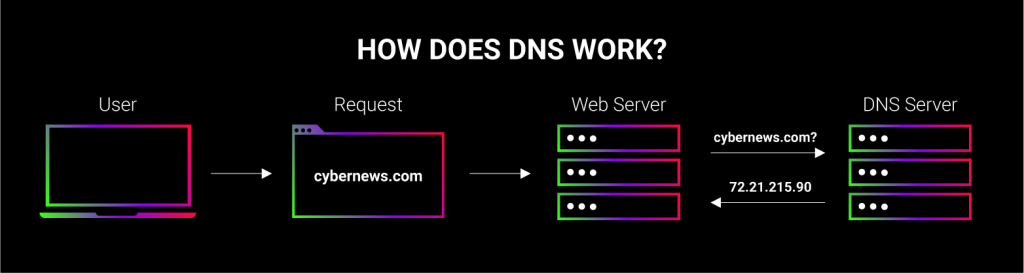
A DNS attack is a cyberattack in which the attacker exploits vulnerabilities in the Domain Name System. This is a grave issue in cybersecurity because the DNS system is a crucial part of the internet infrastructure and at the same time, it has many security holes.

There are many different ways in which DNS can be attacked. DNS reflection attacks, DoS, DDoS, and DNS poisoning are just some of the attack types DNS is susceptible to.

In this article, we'll be discussing **DNS attacks** and **how we can respond to them**.

**What is DNS?**

Before we get into the nuts and bolts of how the attacks happen, let's go through some of the basics of how the DNS or domain name system works.



For the sake of simplicity, think of DNS as a massive phone book that refers to IP addresses with assigned domain names. Your browser doesn't "understand" domain names – to retrieve a website, it needs the IP address of the server where it is hosted. So, when you enter a domain name, this DNS phone book finds the IP to connect to.

### DNS cache

The lookup process wouldn't be very efficient if you had to go through the whole phone book each time you're calling your mom and dad. Likewise, your computer doesn't always have to contact a remote DNS server each time it needs an IP address – for that, it relies on the DNS cache, which is historical DNS lookup information, stored on your browser, OS, router, and at other steps of the DNS lookup process.

If you're trying to visit a site that your closest DNS resolver doesn't know the assigned IP address of, it will ask other DNS servers until it finds the IP. The DNS server then learns of this new site and ads the assigned IP address to the domain name, which is further shared across other DNS servers.

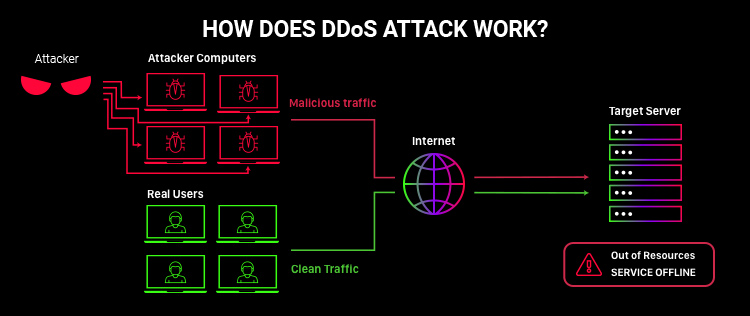
## DNS attack

When hackers take advantage of vulnerabilities in the Domain Name System (DNS), we call this a DNS attack.

Some of the most common types of DNS attacks are the DDoS attack, DNS rebinding attack, cache poisoning, Distributed Reflection DoS attack, DNS Tunneling, DNS hijacking, basic NXDOMAIN attack, Phantom domain attack, Random subdomain attack, TCP SYN Floods, and Domain lock-up attack. We'll look at each of them in this article.

### DoS and DDoS attacks

A[Distributed Denial-of-Service](https://cybernews.com/security/ddos-attacks-explained/) (DDoS) attack is a hostile attempt to interrupt the regular traffic of a targeted network or server by bombarding the network or its surrounding infrastructure with internet traffic. Albeit DDoS isn’t necessarily a DNS attack, the DNS system is a popular target.



DDoS attacks achieve effectiveness by using multiple compromised computer systems as sources of attack traffic. Usually, attackers deploy bots to bombard the target with traffic. A case whereby only one bot is used is known as a Denial Of Service (DoS) attack and is mostly localized or has minimal effect. DDoS, on the other hand, has a more broad impact and will require more resources.

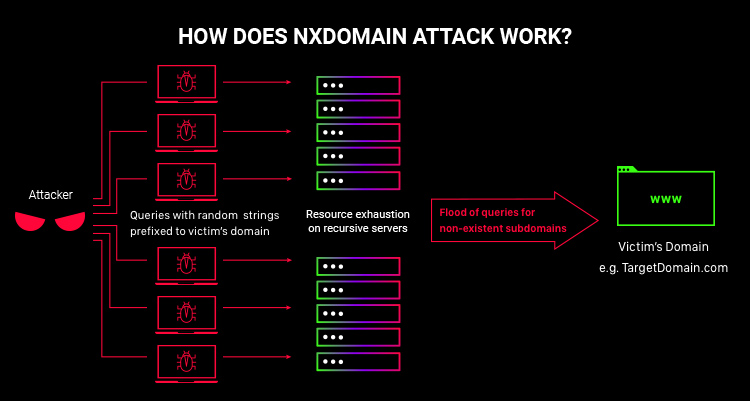
Exploited machines may include computers and other networked resources, such as Internet of Things (IoT) devices. To better understand how a DDoS attack works, imagine a highway artificially clogged up with cars, thereby preventing regular traffic and causing a standstill traffic jam.

There are many types of DDoS attacks aimed at DNS, some of which we will discuss below.

One of the biggest DDoS attacks was the Dyn DNS attack. Dyn is an Internet Performance Management (IPM) company - a pioneering DNS service provider. The Dyn attack occurred on the 21 October 2016. It affected a large portion of the internet in the US and Europe. The source of the attack was the Mirai botnet, consisting of IoT devices such as printers, Internet Protocol (IP) cameras, and digital video recorders.

### NXDOMAIN attack

An NXDOMAIN attack is a DDoS variant when the DNS server is flooded with queries to non-existent domain names, flooding the authoritative name-server’s cache and stopping legitimate DNS requests altogether.



As you already know, your visits to websites are made possible by DNS converting domain names to an IP address. Suppose you’d type in asdasdasdasd.com into your address bar. What would happen then is that the DNS would not find the corresponding IP address because it doesn’t exist and return an error message. However, the resolver still tries to find the result, spending precious milliseconds to look through the cache, using CPU processing power, etc. In other words, before returning the error message, the request was processed along with other genuine requests.

Now, imagine that the attacker controls a botnet containing thousands of users. Each of them sends a request for a domain that doesn’t exist. This could clog the DNS server cache fairly quickly, denying service to users wanting to visit a legitimate site.

In recent times, some Internet Service Providers (ISPs) have started taking advantage of this situation in a harmful way. Instead of returning an error message, they direct these requests to servers with embedded ads, thus capitalizing on the invalid requests.

### Phantom domain attack

A phantom domain attack is a type of DoS attack, directed towards an authoritative nameserver. It is done by setting up a bunch of DNS servers that don’t respond to DNS requests or do it very slowly, interrupting communications.

When a DNS server doesn’t know an IP address, it will look the address up on other connected DNS servers - this is known as recursive DNS. Phantom domain attacks are a method to intercept that lookup process. This wastes the server’s resources on non-functional or inefficient lookups.

When resources are fully consumed, the DNS recursive server may ignore legitimate queries and continue to focus on the non-responsive servers, causing severe performance issues.

### Random subdomain attack

A random subdomain attack is very similar to NXDOMAIN attacks, the difference is that instead of asking the DNS for a non-existent domain, this attack asks for a non-existent subdomain.

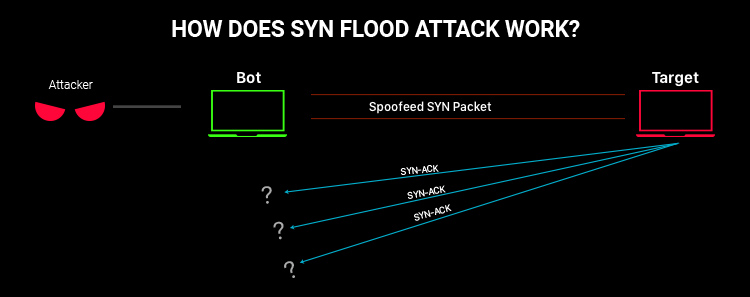
Let us consider this scenario: imagine we want to access www.perfectacademy.org. Since this domain exists, we would definitely get a response to access the Perfect Academy website. If we then remove the "www" part and replace it with a random string, say dhutz.perfectacademy.org, the recursive DNS server will be forced to open a recursive context looking for that "dhutz" string from Perfect Academy's authoritative servers.

This will result in an NXDOMAIN response, that would be stored in the DNS server's negative cache (which is more like a store for non-existent domains). If the "dhutz" label was changed continuously, then each query would trigger a recursive query to Perfect Academy's authoritative servers, consuming recursive contexts and populating the negative cache.

In effect, NXDOMAIN is much broader in scope and scale. Meanwhile, the random subdomain attack targets the domain’s authoritative nameservers in particular.

### TCP SYN floods

Transmission Control Protocol Synchronize (TCP SYN) flood attack is a form of DDoS attack that interrupts the handshake between the server and client by flooding it with arbitrary requests.



Rather than exhausting the server's processing power, this attack aims to exhaust the reserve of available open connections. It achieves this by sending bursts of synchronize (SYN) messages to the server faster than it can respond to them. A typical three-way handshake simply involves the client sending a synchronize (SYN) message to the server, the server responds with a synchronize-acknowledge (SYN-ACK) message. While the server is preparing a SYN-ACK message as a reply, the attacker generates more and more requests,

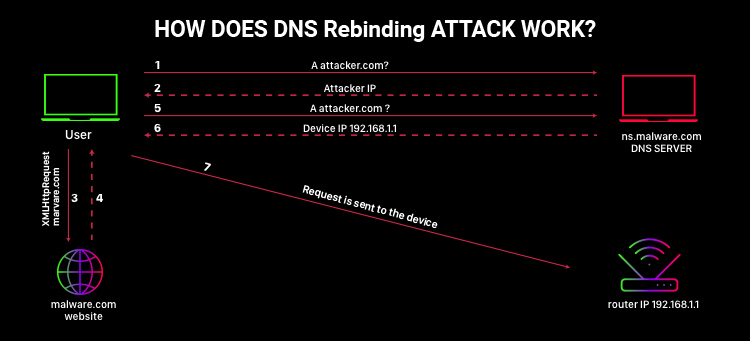
### DNS domain lock-up attack

The DNS domain lock-up attack is a form of DDoS attack with specially set up special domains and resolvers that also interrupt the handshake between the server and the client by not sending out the correct response but by replying with random data packets. They keep the server engaged and waiting for a proper reply (which never comes) exhausting the reserve of available connections.

The main difference between this and the TCP SYN flood is that the DNS domain lock-up attack happens in the next phase of a three-way TCP handshake. To successfully establish a connection, the client sends out a SYN message, the server replies with a SYN-ACK message and waits for an ACK message back from the client. DNS domain lock-up attack deliberately slows down the handshake, sending back ACK messages from the attacker-side. These false domains respond by sending a random or useless packet data to keep the DNS resolver occupied, unable to resolve the handshake. This completely negates all other legitimate connections for actual users.

### DNS rebinding attack

DNS rebinding attacks use DNS vulnerabilities to bypass the web browser’s same-origin policy, allowing one domain to make requests to another - something that can have far-reaching consequences. For example, using DNS rebinding, an attacker may be able to gain control of your entire home network.



Picture this: you're browsing a shady website, which happens to have a malicious script running: <script src="http://clear-your-bank-account.com/ad.js">.

For protection, the script will typically only be able to access the domain you are currently browsing and not some other domain (such as your-bank.com) because of the same-origin policy. This is one of the most essential safety measures of the internet, and all browsers enforce this. It ensures that a malicious script running on one website will not be able to send requests to another website, and thus won’t be able to, for example, clear your bank account.

This, however, is very much exploitable using DNS rebinding.

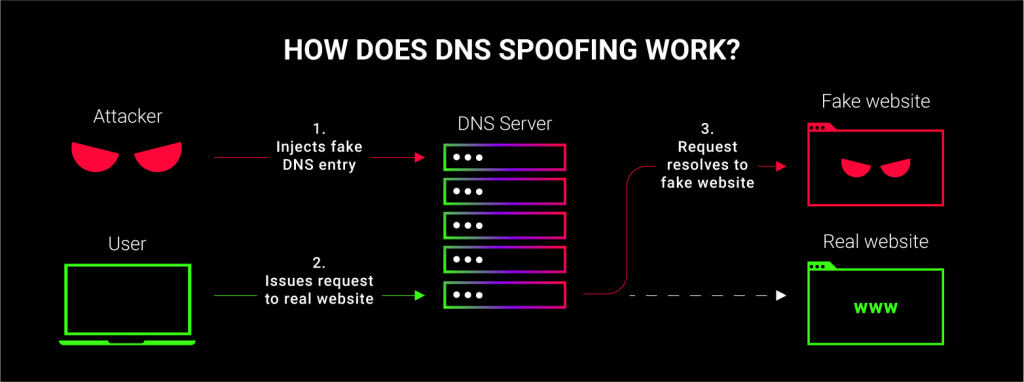
When performing a DNS rebinding attack, the hacker registers a web domain, i.e., malware.com, and assigns it to its own DNS server, giving the lookup response a very short time to live (TTL) to prevent DNS caching and forcing your browser to perform repeated lookups. The attacker then gets his victim to load malware.com on their browser (this can be done via phishing or a number of other means). When the victim loads the website, it triggers the malicious script on site.

This is where it gets interesting: the script starts making weird requests, which will depend on the attacker’s goal. It’s not a problem if the requests only get as far as malware.com. However, since the set TTL time is very low, another DNS lookup is performed, only now the response is a different IP address - the victim’s home router, for example.

The reason this works is the DNS links different IP addresses to the same domain name, thus bypassing the browser’s same origin policy.

### DNS cache poisoning, a.k.a DNS poisoning

DNS cache poisoning is something that happens when there are incorrect IP addresses stored on a DNS cache. For example, instead of leading a user to amazon.com, the incorrect DNS cache entry might lead users to a phishing website that looks like the Amazon website. DNS poisoning can happen by design, because DNS servers rely on each other to answer lookup queries, allowing misinformation to spread.



The way DNS poisoning attacks typically happen is this:

* the attackers impersonate a DNS name server
* they make a request to a DNS resolver
* they forge a reply to the DNS resolver before the real DNS name server can answer

DNS requests and queries use UDP (User Datagram Protocol), which doesn't require a handshake to verify that the recipient is who they claim to be. Through this UDP vulnerability, the attacker can send a forged response with false header data that will route a connection somewhere else.

Since there is no way to check whether the entry is genuine or not, the DNS resolver automatically caches the data. This means the cache is now poisoned and it will stay poisoned until the entry’s time to live (TTL) expires, or the DNS cache is manually flushed.

Every time the user will try to enter some web address the attackers have tampered with, your browser will retrieve the incorrect address from the cache because it's faster.

Despite seemingly built-in security vulnerabilities in the DNS caching process, DNS poisoning attacks aren't easy. To get the cache poisoned, the attacker has a very short timeframe to get in the middle and send a fake reply before the actual response from the correct nameserver comes back.

On top of that, to successfully spoof the users, the attackers need to know several external factors. For example, a DNS resolver may use randomized ports, request ID number, the actual nameserver the query goes to, etc. Without this information, the attack won't be successful.

**How to mitigate a DNS attack**

Now we understand that attackers are not super hackers that cannot be stopped. All they do is just look for vulnerabilities in the DNS and attack them

There are a few things we can do as users to mitigate attacks on DNS:

1. If you operate your own DNS resolver, restrict the usage to only users connected to your network. This will help to prevent attackers from poisoning your resolver's cache.
2. If you run your own DNS server, then make sure you keep the DNS server and the OS they run patched and updated to prevent them from being exploited due to known vulnerabilities.

If you use a domain name registrar, you can also protect yourself from DNS attacks:

1. DNSSEC allows DNS data to be digitally signed so that it becomes impossible for an attacker to forge it. So be sure to confirm if your provider has implemented DNSSEC.
2. Make use of two-factor authentication. If attackers gain access to one of your administrator's account details, two-factor authentication will still make your DNS safe because access to the account will depend on a second authentication factor such as a one-time password sent to a mobile phone or email address.
3. You should enable modification locking. This feature requires a specific action to be performed before any change can be made

**What is DNSSEC?**

Domain Name System Security Extensions creates a secure DNS by attaching cryptographic signatures to already available DNS records. These [digital signatures](https://cybernews.com/privacy/digital-signature-guide/) are kept in DNS name servers with regular record types like Mail Exchanger (MX), Canonical Name (CNAME), and so on.

By checking its associated signature, you can verify that a requested DNS record came from its authoritative name server and was not altered while in transmission, as opposed to a fake record injected in a man-in-the-middle attack.

Since DNSSEC is an extension of DNS, it adds a few new DNS record types such as:

1. RRset Signature (RRSIG), which contains zones with a group of records of the same type
2. DNSKEY, which contains a public signing key.
3. Delegation Signer (DS) containing the hash (like a reference) of a DNSKEY record. This allows the transfer of trust from a parent zone to a child zone. DS helps resolvers ascertain that a child is authenticated
4. Next Secure (NSEC) and NSEC3 which is used by resolvers to ascertain the non-existence of a record name and type. Usually, an NXDOMAIN response occurs when a non-existent domain is requested. Still, instead of this response, a "next secure" record is returned instead, which provides a result (domain) close to the one queried or requested.

### DNSSEC tools

To successfully deploy DNSSEC on the client and server, you will need to install special software.

Some of the software tools needed are:

1. Windows 7 and Windows Server 2008 R2: it includes a "security-aware" stub resolver that can distinguish between secure and spam responses by a recursive name server.
2. Windows Server 2012 DNSSEC: it's compatible with secure dynamic updates with Active Directory-integrated zones.
3. BIND: this incorporates the newer DNSSEC-bis (DS records) protocol as well as support for NSEC3 records.
4. Unbound: this is a DNS name server completely written from scratch with DNSSEC concepts in mind. Other examples include mysqlBind, OpenDNSSEC, Knot DNS, PowerDNS