WHITE PAPER

THREATS TO DNS ARE THREATS TO YOUR BUSINESS

What You Can Do

neustar®

Security Services

TABLE OF CONTENTS

Introduction

The Domain Name System (DNS) is among the foundational elements of the internet, connecting users to online assets by easy-to-remember domain names, rather than by IP addresses. DNS as we know it was invented in 1983, taking the system into broad use outside of the academic community. As the internet has grown, DNS has grown too, becoming essential in a world where online assets could be diversified across various platforms or locations in order to better serve their audiences. DNS makes it possible for users to go beyond just getting a result to a query—to getting the best result, fastest response, and more. When it is implemented well, DNS can become a robust tool for business, particularly as the world has moved online.

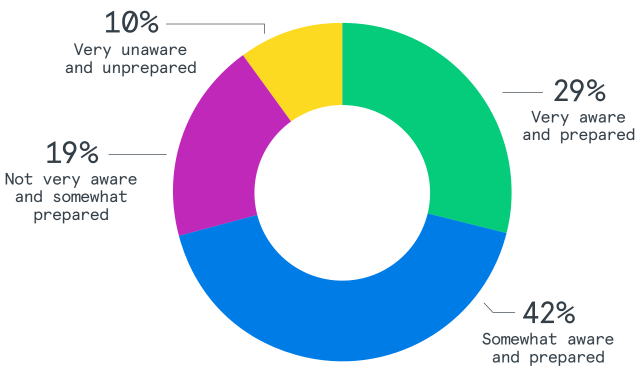
However, DNS is architected in such a way that it's easy to take the system's "it just works" functionality for granted, and organizations increasingly do so at their peril. Attacks on DNS are being enacted every day. While these attacks may not take major swaths of the internet offline, they remain extremely costly, especially with the push to digital transformation due to the COVID-19 pandemic. According to the International Data Corporation (IDC), the average cost to an organization as a result of a DNS attack in 2020 was $924K, mostly due to application downtime.[1]

Why Is That Important?

Some companies manage their own DNS. Others may choose to have their DNS managed by their ISP/registrar, and others prefer a third-party service. Regardless of which method is chosen, there are dangers ahead—according to IDC, 79 percent of the organizations responding to their 2020 survey had suffered from a DNS attack.[2] Statistically if your organization has not experienced such a threat yet, it is likely that you will. And when you do experience an attack, it is vital to have a plan; unfortunately, many organizations do not have confidence in their ability to respond to attacks, as shown in response to the question asked in the November 2020 Neustar International Security Council (NISC) Survey.

ORGANIZATION'S AWARENESS AND PREPAREDNESS TO RESPOND TO THREATS LAUNCHED AS PART OF A DNS ATTACK

Around 3 in 10 admit to having some reservations about their organization's awareness and preparedness to respond to DNS attack threats.



Answers to organizations preparedness to respond to DNS attacks, NISC Survey, November 2020

DNS Attacks

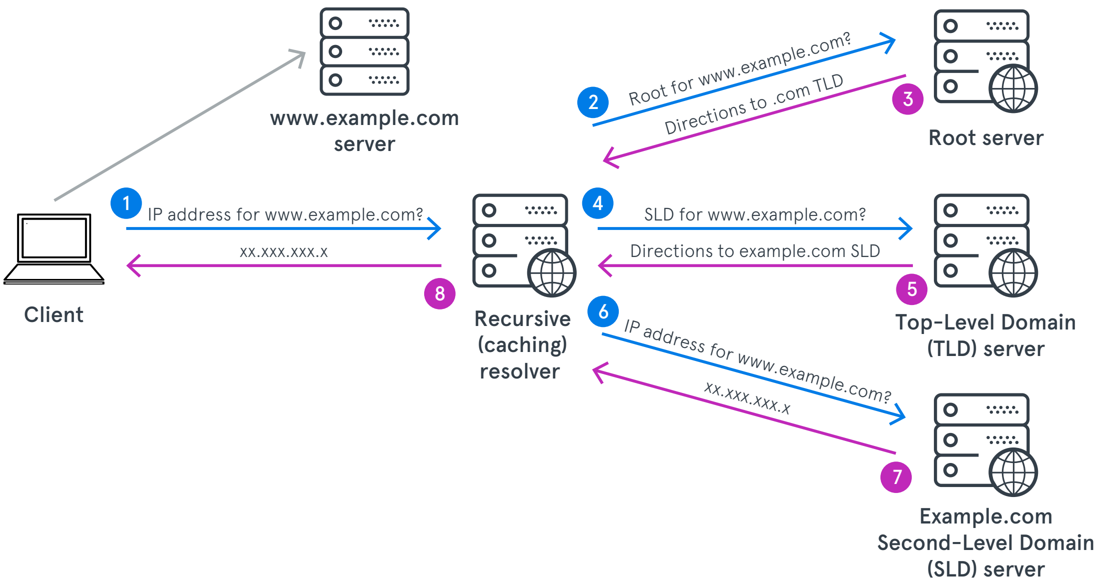
DNS is a foundational part of the internet, and its hierarchal structure is well known and understood.

However, DNS is much more than a single point product/set of products. DNS touches upon a myriad of other technologies that are critical to operating your business. There are many interconnected operations within the address resolution process, and bad actors can inject themselves at various stages, making the definitions of discrete attack types unclear. It is often more useful to consider the attacker's end goal, as opposed to looking at threats to DNS by named attack types, to clarify issues and avoid confusion.

Primary categories of DNS attacks include:

- Attacks that deliver a bad/false answer to a DNS query, such as cache poisoning, domain hijacking, man-in-the-middle, and other exploits.
- Attacks that prevent requestors from getting any answer to a query, including distributed denial-of-service attacks.
- Attacks that use DNS as a transport mechanism, including those that bypass firewalls, access control lists (ACLs), intrusion detection systems (IDS), and more.

DNS HIERARCHICAL STRUCTURE


www.example.com server
2
Root for www.example.com?
Directions to .com TLD
3
Root server
1
IP address for www.example.com?
xx.xxx.xxx.x
8
Client
Recursive (caching) resolver
4
SLD for www.example.com?
Directions to example.com SLD
5
Top-Level Domain (TLD) server
6
IP address for www.example.com?
xx.xxx.xxx.x
7
Example.com Second-Level Domain (SLD) server

CATEGORY 1

Attacks that Deliver a Bad/ False Answer to DNS Queries

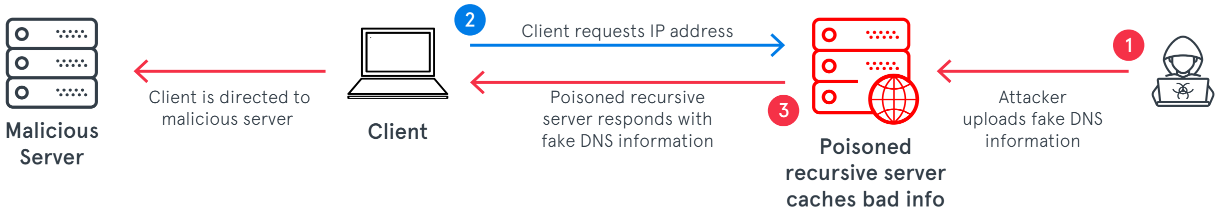
The goal of this category of attack is to redirect users by sending them to a malicious site rather than to the actual site they were seeking, or to even black-hole their request altogether. When successful, this exploit allows the attacker to effectively take down an online presence or impersonate the legitimate site. Impersonation can lead to disclosure of user login credentials, propagation of false information, infection with malware or ransomware, and the ability to observe and/or eavesdrop on emails. Different methods to accomplish the intended goal can include cache poisoning, false authoritative DNS servers, domain/DNS hijacking, DNS zone modification, and more. While each of these attacks have the same goal, they each target different elements within a DNS query and the overall infrastructure.

Cache Poisoning

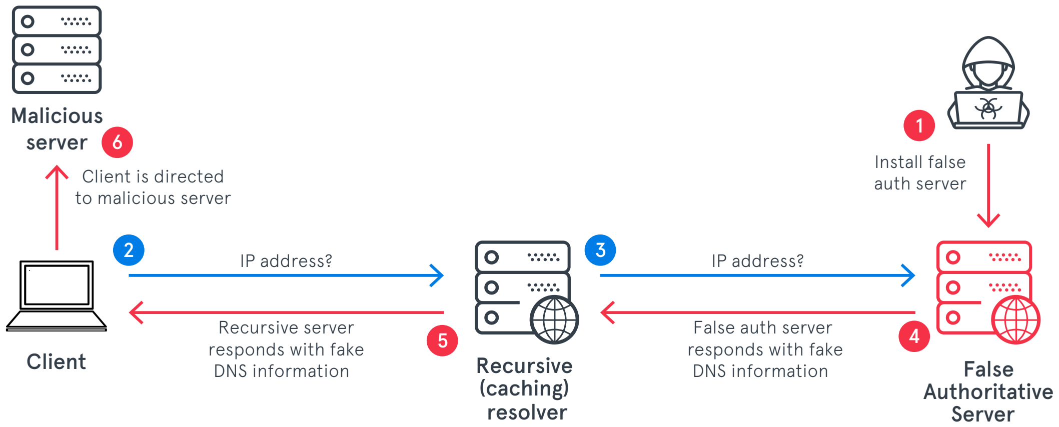
This attack seeks to replace the legitimate information that is cached in recursive resolvers with false data. When successful, this attack injects bad address data at the beginning of the DNS address resolution process. Because the recursive server has an answer to the query in its cache, it will not forward the request to the real authoritative server for the correct information. The compromised, or "poisoned," recursive resolver will simply continue to hand out the "bad" answer that has been injected into its cache until the reference times out.

Cache poisoning attacks hinge upon the fact that DNS, by default, runs over User Datagram Protocol (UDP). UDP is a connectionless protocol designed for delivery speed rather than error-free, guaranteed delivery transmissions. That means if a resolver asks for an address to be resolved and receives an answer via UDP, be it good or bad, that answer, if timed correctly with a related query, could end up being cached. Cache poisoning is a brute force attack, in which hundreds of requests are sent at the same time that hundreds of bad address "resolutions" are sent. Using this method, there is a chance that the bad address might be cached. The bad responses usually include a very long Time to Live (TTL) to give them staying power within the "poisoned" recursive resolver. The timing to make this attack successful is tricky, but, as researcher Dan Kaminsky showed first in 2008 and again in 2020, it is not impossible.

CACHE POISONING ATTACK


Malicious Server
Client is directed to malicious server
Client
2
Client requests IP address
Poisoned recursive server responds with fake DNS information
3
Poisoned recursive server caches bad info
1
Attacker uploads fake DNS information

FALSE AUTHORITATIVE SERVER


Malicious server
6
Client is directed to malicious server
1
Install false auth server
2
IP address?
3
IP address?
Client
Recursive server responds with fake DNS information
5
False auth server responds with fake DNS information
4
Recursive (caching) resolver
False Authoritative Server

False Authoritative Servers

Authoritative servers are designed to return an IP address in response to queries from recursive resolvers. If an attacker can manage to direct queries to a false authoritative nameserver, via modification of a legitimate DNS zone file, domain hijack, or other exploit, they can direct users to malicious sites under the auspices of correct name resolution. Such attacks can enable bad guys to reroute users and to change where web queries or emails go, among other things. Setting up a false authoritative server could also be accomplished by domain hijacking or DNS zone modifications.

What to Do

Luckily there is a means to defend against these attacks and others like them—Domain Name System Security Extension or DNSSEC. If a domain is DNSSEC-enabled, in the case of cache poisoning, the recursive resolver will be able to validate that the response to a query is legitimate because DNSSEC is enabled throughout the DNS hierarchy. In the case of false authoritative servers, the attacker will not have the appropriate key component to successfully complete the DNSSEC validation. That means that the false authoritative servers cannot provide the appropriate response to the DNSSEC request, and as a result, they won't be accepted by the recursive resolver.

Another consideration is to ensure that your DNS vendor is in a position to ensure transitive trust. A good place to start is to work with a vendor that does not subcontract with third parties to offload DNS resolution. If that is not possible, investigate any subcontracted organizations as if your contract were with them directly—because when it comes to DNS query resolution, it may very well be.

DNS from the Inside Out

The attacks above happen at various points in the address resolution process, and bad addresses are injected at various points from the outside. Another way to reach the same goal targets DNS from the inside. This type of exploit can be used to direct users to a bad site and can also disable address resolution completely. When successful, the attacker can then enjoy all of the same activities outlined earlier, including gathering credentials/information, feeding the user erroneous data and/or malware, or eavesdropping on communications. These attacks can happen in several places in the DNS hierarchy.

DNSSEC: What's in a name?

DNSSEC is a method of a validating the legitimacy of a DNS address resolution and provides end-to-end data integrity checks. The protocol uses asymmetric encryption, also known as public key cryptography. This encryption method, which is also used in SSL/TLS encryption, is based on two different but mathematically related keys that are used to encrypt and decrypt transmissions. One is private and kept securely hidden, while one is public.

DNSSEC uses encrypted digital signatures rather than encrypting the traffic, as SSL does. Once an owner digitally signs their zone, the public key is included with the other zone data. Query results validated with DNSSEC are therefore ensured to be the correct and unmodified responses. If the signatures do not validate, then the request will fail, and the user will not be sent to a bad site.

One essential aspect of DNSSEC is the fact that the entire DNS hierarchy you're using needs to be DNSSEC-enabled for it to function. This means that your top-level domain (TLD)—such as .com, .biz, or .org, to name a few—must be signed, and to date, the vast majority of common TLDs are. Many—but not all—country code TLDs (ccTLDs) have also been signed, but regardless of your TLD, it is important to check. If you have a question about whether your domain supports DNSSEC, the Internet Corporations for Assigned Names and Numbers (ICANN) maintains a current list here.

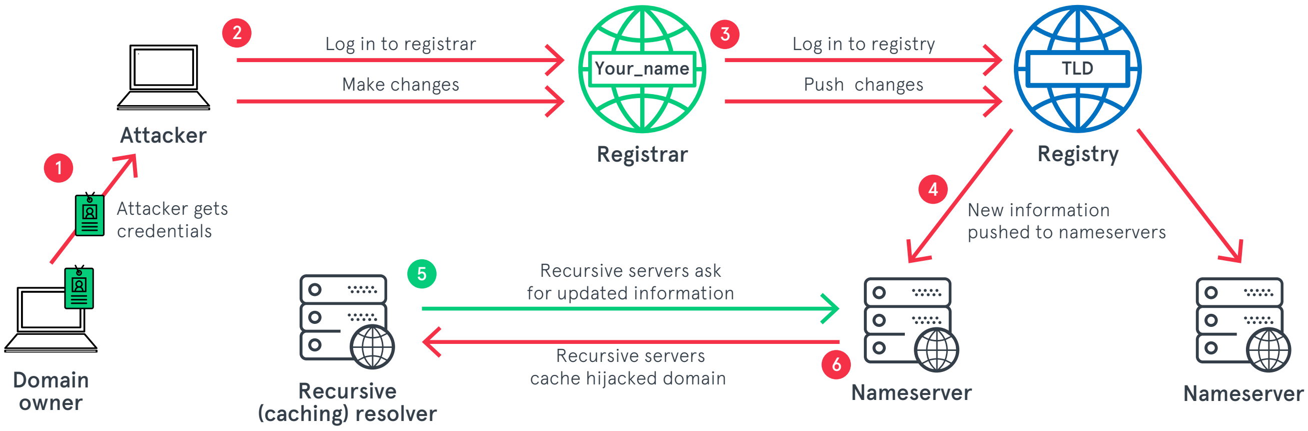
Your domain registrar must also support DNSSEC, as they need to be able to accept and sign delegation signer (DS) records and include them in the parent domain. Next, the DNS operator, which could be the registrar, the owner, or a third party, must support DNSSEC. Finally, the zone owner must explicitly enable DNSSEC signing of the zone.

Domain/DNS Hijacking

In a DNS/domain hijacking attack, the bad actor somehow works their way into the position of acting as the domain owner in order to make changes, just as the owner would. The phrase "as the owner would" is the key to this exploit, as these attacks are nearly always either inside jobs, theft of login credentials, or the result of successful social engineering or phishing.

DNS hijacking attacks look to the outsider as if the system is acting normally. The attacker impersonating the zone owner logs in to the registrar and pushes some changes, the registrar exchanges data with the registry, and the registry then pushes the changes to the appropriate nameservers. One of the best-known DNS hijacking attacks on record happened in 2017, when hackers took over an entire Brazilian bank's online footprint for five hours. The attackers got into the bank's account at the registrar for that TLD, who was also acting at the bank's DNS operator. The attackers simultaneously changed the registration for all of the bank's online properties, redirecting them to servers that the bad guys had provisioned in the cloud. Anyone visiting the bank's URLs during that period were sent to look-alike sites that had been built to simulate the bank's own sites, right down to valid SSL certificates. In this exploit, the attackers did not rob the bank; they became the bank. The hijack not only took over the public website of the bank, but also redirected control over their email servers, so there was almost no way for the bank to warn customers of the compromise. Not content to merely phish for credentials, the hijacked sites also installed malware designed to look like an update to the bank's trusted communication software. This malware harvested logins from a variety of other banks, as well as email information, File Transfer Protocol (FTP) credentials, and contact lists from Outlook and Exchange. Similar incidents have occurred since this takeover.

DNS/DOMAIN HIJACKING


2
Log in to registrar
Make changes
Your_name
Registrar
3
Log in to registry
Push changes
TLD
Registry
Attacker
1
Attacker gets credentials
Domain owner
4
New information pushed to nameservers
5
Recursive servers ask for updated information
Recursive servers cache hijacked domain
6
Recursive (caching) resolver
Nameserver
Nameserver

What You Can Do

Just like anything else you are trying to safeguard, the best idea to secure your domains is to lock the points of entry. In this case, it means providing another layer of interaction between the domain owner and the registry and registrar, in addition to keeping careful track of the use of login credentials. Not all registries/registrars offer these locks, so be sure to check for them and use them if available.

- **Registrar locks:** This service requires the registrar to confirm any requested changes with the domain owner. In the case of urgent cries for help (which may come from the attacker, trying to scam the registrar), the lock demands that the domain owner validate the transaction in one of a number of ways to include two-factor authentication and requiring the owner to provide a passphrase.
- **Registry locks:** Registry locks are a more stringent, manual (and sometimes offline) process that effectively neutralizes any attempts by fraudsters to social engineer your domain registrar.

With a registry lock in place, modifications of any kind are prohibited without the domain owner's validation of the change. Among other things, your registrar cannot move your domain to another registrar on its own—a manual contact verification by the corresponding registry is required.

Other actions that you can and should take are to follow general security best practices, including:

- Protect your account credentials!
- Use multi-factor authentication and require its use by all users and subcontractors.
- Keep track of all contact and recovery emails to make sure they are company controlled, not personal emails.
- Review existing accounts with registrars and others.
- Ensure that you have notifications in place about expiration dates.
- Monitor the issuance of any new SSL certificates for your domains.

CATEGORY 2

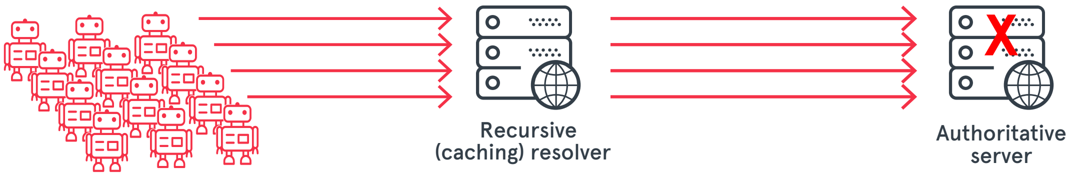
Attacks that Prevent Any Answer to DNS Queries

While redirecting users to malicious sites is a goal for some attackers, others would prefer to take a particular site offline completely. One way to do that is via a distributed denial-of-service (DDoS) attack directed against the target's DNS infrastructure. After all, if a domain name won't resolve, users cannot reach the site. Because DNS must be open to the internet to function, it can be a compelling DDoS target.

Like other DDoS attacks, there are several methods that can be leveraged to make DNS unavailable, including volumetric and state exhaustion attacks. Volumetric attacks seek to fill all available bandwidth and are generally measured in bits (or gigabits) per second. State exhaustion attacks, on the other hand, seek to max out the operating capacity of an element of the infrastructure.

There are a variety of DDoS attacks that are regularly directed toward DNS infrastructure. Many of these attacks use botnets to generate a sufficient volume of requests, but not necessarily to clog bandwidth. Most attacks today attempt to use up the resources of the devices in the DNS infrastructure instead. One DDoS tactic, the DNS flood, comprises large volumes of malformed packets. These attacks seek to consume server-side resources and can exhaust capacity for both recursive and authoritative servers. Another exploit, the NXDOMAIN or Phantom Domain attack, involves a botnet sending requests for subdomains that do not exist. Because they cannot find the answer in cache, resolvers will forward these requests to the authoritative server. Either the recursive or authoritative servers can be overwhelmed in such an attack. It is likely that authoritative servers will fall first, as they are only provisioned to take a percentage of the traffic handled by recursive servers.

NXDOMAIN ATTACK



What You Can Do

- Ensure that your DNS infrastructure is over-provisioned, although the use of this term begs the question of how much excess capacity is really necessary. Older best practice documents used to recommend provisioning DNS for 10x the expected load, but given the rise of IoT devices and associated botnets, that number is simply not adequate to prevent the sort of threats we've seen in recent years. Resolution infrastructure should be 100x to handle the ebbs and flows of internet traffic, as well as the potential for attack. Neustar Security's DNS infrastructure, for example, is currently provisioned at well over 100x query volume.
- Make sure that your DNS infrastructure is protected by robust DDoS mitigation. In today's climate, DDoS mitigation is more than a nice-to-have feature. DDoS is a constant threat to DNS, and the worst time to consider mitigation is when you are under attack. Regardless of the company you choose, DDoS mitigation is possibly a more essential protection for your DNS than anywhere else, as an attack will deny users any access to your company or applications.

DNS as a DDoS Amplification Vector

In addition to being an attack target, DNS can be used as an attack vector—used to generate DDoS attacks on other DNS infrastructures or on other services. To use DNS as an attack vector, the attacker creates a request using a spoofed source address of the victim being targeted, which ensures that answers to the query will be returned to the victim's IP address. The query, which is designed to elicit a large answer (thus amplifying the attack), is sent to many open recursive DNS resolvers. The responses from these resolvers are sent to the victim, flooding them with DNS answers that they did not request. Since a DNS query of as small as 60 to 80 bytes can generate a DNS response many times larger, the floods can reach terabytes, all headed for the intended victim.

What You Can Do

Given the ratio of over-provisioning that is recommended for enterprise-grade DNS infrastructure, such an attack may not greatly impact the organization being used as an unwitting accomplice, but security best practices can help you avoid the issue completely.

- Close any "open" DNS recursive servers in your network. By restricting your recursive servers to internal IP addresses only, you can easily ensure that they cannot be used to facilitate an attack.
- Consider rate-limiting responses from your authoritative nameservers.

Neustar Security Services has always recognized the importance of DDoS protection for DNS. In fact, our market-leading UltraDDoS Protect service was originally built to provide security for UltraDNS. Today, UltraDDoS Protect has grown to over 12+ Tbps of scrubbing capability, with one of the largest offerings in the world. UltraDDoS Protect provides always-on and on-demand DDoS mitigation for a diverse set of global customers, securing everything from cloud-based resources and websites to enterprise infrastructure. Many UltraDDoS Protect scrubbing centers are co-located with UltraDNS, so defense is always close at hand.

CATEGORY 3

Attacks that Use DNS as a Transport Mechanism

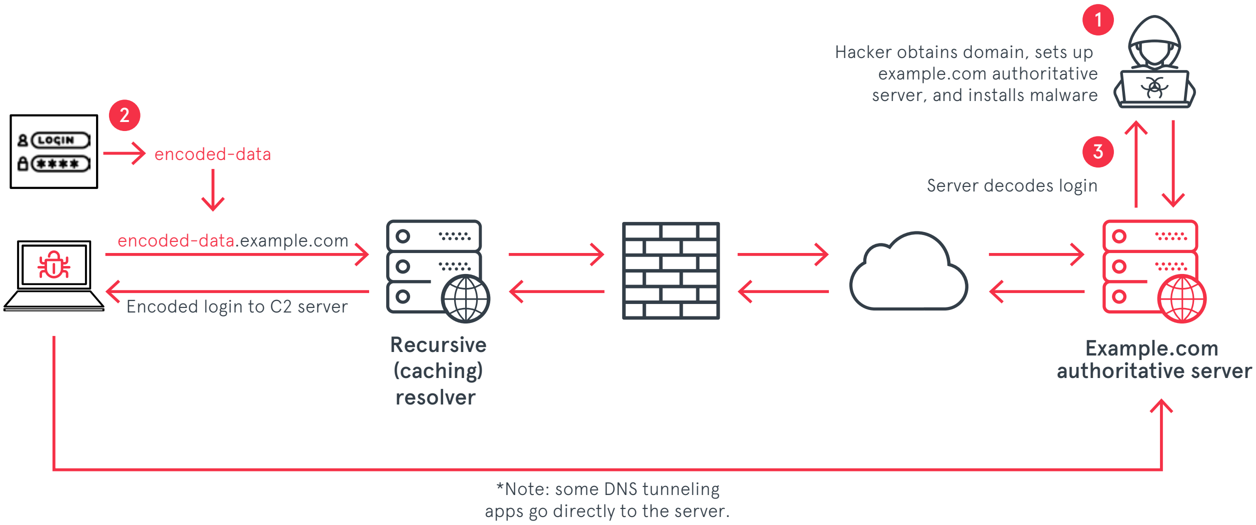
The previous sections in this paper have looked at ways to use DNS to slow or halt access to a legitimate site or to redirect traffic to a malicious site. In these scenarios, the attacks focus on DNS doing its job to resolve domains into IP addresses. DNS tunneling, however, is a fundamentally different way that attackers can make use of the ubiquity of the system.

DNS tunneling uses the DNS traffic as a transport mechanism to transmit packets into or out of the enterprise. The danger of this attack comes from the fact that DNS is not typically used for data transfer, so many companies do not monitor DNS traffic for malicious activity as they would with other traffic types. DNS tunneling can be used as a command and control (C2) channel for malware as well as a means to exfiltrate data or pass other IP traffic. Ironically, many of the tools for staging this attack were originally created to bypass the captive portals of paid Wi-Fi services. In that use case, if a Wi-Fi service allowed DNS traffic to traverse the network, a tunnel could be set up to send traffic encoded in the DNS query and response packets without having to pay for the use of the Wi-Fi service. The exploit has been around for over two decades and remains pretty much as originally conceived. Even though free Wi-Fi has become the norm these days, the method remains intact and is now being used to smuggle malware/commands into and data or other traffic out.

It is easy to find toolkits that enable setup for DNS tunneling. The typical infrastructure includes a system inside the LAN with the appropriate software installed—bear in mind that this software may have been "installed" via a malware infection! Next, the attacker needs a domain name for the client/malware to query for in order to enable the DNS traffic and an authoritative server outside the LAN to backstop the communication.

For this example, we'll assume that the "client" inside the LAN is actually malware on a user's device. The malware can be configured to periodically check in with its command and control (C2) server for new instructions. The malware encodes data, such as login credentials to the hacker's server, then puts the payload into a DNS query to the hacker's domain. The request will pass through the network to the hacker's "authoritative" server, where the message is decoded and read. The attacker then encodes the reply in the fields of the DNS response. The attacker can essentially operate the client's device remotely and use that access to infect other systems inside the firewall or to exfiltrate data.

DNS TUNNELING


1
Hacker obtains domain, sets up example.com authoritative server, and installs malware
2
LOGIN
****
encoded-data
encoded-data.example.com
Encoded login to C2 server
Recursive (caching) resolver
3
Server decodes login
Example.com authoritative server
*Note: some DNS tunneling apps go directly to the server.

What You Can Do

The most important thing that you can do to stop DNS tunneling is to be aware that it exists. Running secure recursive resolvers inside the network and fronting them with DNS firewalls could work if and when the bad address is known. It may be appropriate to set your firewall to block all outbound port 53 traffic from any machine inside the firewall (other than the company blessed recursive servers), as authoritative servers will bind to that port.

The key is to identify anomalous traffic operating under the cover of DNS. To do that, you need to closely monitor things like the length of subdomain names in queries, as these could be encoded data. Another thing to look for is DNS requests with high entropy or a high level of disorganization. Most DNS requests follow a fairly organized pattern. Because DNS traffic is passed in the clear by default, it is possible to spot traffic that doesn't look similar to others. A current threat feed, such as Neustar Security Services' UltraThreat Feeds, will include information gleaned from a variety of sources and can be used to keep your SIEM or other perimeter security devices up to date. It is also important to keep a close eye on the domains themselves as well as the frequency and source of specific requests over time.

Protecting Your DNS = Protecting Your Business

The first step in providing DNS protection is to realize the fact that DNS is central to operations; after all, virtually every communication on the internet begins there. You can ultimately save both time and money by considering how essential DNS is to your business and taking steps to protect it proactively, rather than having to scramble to recover in reaction to an exploit.

Consider the cost of legitimate users turning away from your site because it is slow to load; after all, recent research shows that a 2-second delay in load time resulted in abandonment rates of up to 87 percent.[3] What if users are not able to access your site at all? Will they come back? Even worse is the possibility of users being redirected to a malicious site and losing their valuable information or being infected with malware. Any of these scenarios could play out in the case of cache poisoning, DNS/domain hijacking, zone modifications, or DNS DDoS. And, as we've outlined with DNS tunneling, users are not the only entities that could lose information. Unsecured DNS can be used to remove valuable customer information and more.

Now that you've looked at some of the most common DNS attacks, here are a few steps that you can take to ensure that your organization does not fall victim.

- Make sure to use good account hygiene and access controls based on job roles for all DNS related accounts.
- Implement DNSSEC: it is the only way to ensure that the site information returned from a query is legitimate.
- Make sure that there is adequate capacity and protection in the case of a DDoS attack against your DNS infrastructure. As in many other cases, your site does not need to be taken offline to be affected—just slowing down resolution could lose you a chunk of your user base.
- Consider blocking all client 53 traffic other than from sanctioned and monitored recursive servers.

Now for Some Good News

While protecting your DNS does take some effort, the benefits that can be delivered by a well-implemented system are enormous. DNS can ensure that your users get the fastest response possible, without having to change your network or infrastructure. It can enable you to "route around" problems or slow paths, with dynamic features that can failover in the case of timeouts. It can keep users from bad sites and protect them from the danger of known malware. DNS can deliver users to the best content possible, whether that means presenting locale-specific sites or using the closest assets for better response.

Some simple steps can make your DNS a positive business tool while at the same time eliminating some very real risks.

[1] https://www.helpnetsecurity.com/2020/06/11/average-cost-of-dns-attacks/

[2] https://www.helpnetsecurity.com/2020/06/11/average-cost-of-dns-attacks/

[3] https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/

GLOSSARY

ACL – Access Control List

C2 – Command and Control

ccTLD – Country Code Top-Level Domain

DDoS – Distributed Denial of Service

DNS – Domain Name System

DNSSEC – Domain Name System Security Extension

DS – Delegation Signer

FTP – File Transfer Protocol

ICANN – Internet Corporations for Assigned Names and Numbers

IDC – International Data Corporation

IDS – Intrusion Detection System

IP – Internet Protocol

ISP – Internet Service Provider

LAN – Local Area Network

NISC – Neustar International Security Council

NXDOMAIN – Non-existent Domain (i.e. "the answer to your question doesn't exist")

SIEM – Security Information and Event Management

SLD – Second-Level Domain

SSL – Secure Sockets Layer

TLD – Top-Level Domain (i.e. .com, .biz)

TTL – Time to Live

UDP – User Data Protocol

URL – Uniform Resource Locator

About Neustar Security Services

The world’s top brands depend on Neustar Security Services to safeguard their digital infrastructure and online presence. Neustar Security Services offers a suite of cloud-delivered services that are always secure, reliable, and available and enable global businesses to thrive online. The company’s Ultra Secure suite of solutions protects organizations’ networks and applications against risks and downtime, ensuring that businesses and their customers enjoy exceptional [and uninterrupted] interactions all day, every day. Delivering the industry’s best performance and always-on service, Neustar Security Services’ mission-critical security portfolio provides best-in-class DNS, application and network security including DDoS, WAF and Bot management to its global 5000 customers and beyond.

Find more information at:

neustarsecurityservices.com