ARTICLE

DNS Is Still the Achilles' Heel of the Internet

By: Ray Pompon
Date: March 10, 2017

Imagine proposing a new application project to your boss. It's a distributed network database that runs across millions of nodes on the Internet. Everyone would own and run their own server but would need to coordinate data storage, retrieval, and update with all the others.

This cooperation would be based on a published document describing the relationship—but that's all! There would be no organization and no master control server in charge, just some simple hierarchies and some registered authorities of who holds what records. Anyone could query this database anonymously, and the whole distributed system would work out the answer and return it to the requestor. Oh, and the whole thing would run over the fire-and-forget, unreliable User Datagram Protocol (UDP)[i], which can be easily spoofed.

Your boss would probably laugh you out of the room for proposing such an unworkable system. Yet, in 1983, the Internet Engineering Task Force proposed a solution[ii] and the following year, the first Domain Name System (DNS) server was coded at UC Berkley[iii].

This was back in the days when everyone on the net (called ARPANET back then) trusted each other completely and none of the participants were motivated to cause problems. Somehow, good old DNS survived this sheltered childhood and thrives today in our modern swamp of vipers and leeches that is the Internet. It hasn't been without some scars, as DNS still bears some fundamental weaknesses that are still exploited today.

Why has DNS survived? First, it's extremely useful. As you may already know, the job of DNS is to map a name to an Internet numeric address. It's far easier to remember f5labs.com than 165.160.13.20. Even better, you can map multiple numbers to a single name to achieve scalability and reliability. Email services can also be advertised for a domain, so that you know that you can send mail to me (rpompon@f5.com) via:

```
$ nslookup
> set type=mx
> f5.com
Non-authoritative answer:
f5.com       mail exchanger = 20 mail.f5.com.
f5.com       mail exchanger = 30 mail2.f5.com.
f5.com       mail exchanger = 10 mail13.f5.com.
f5.com       mail exchanger = 10 mail15.f5.com.

> mail.f5.com
Non-authoritative answer:
Name: mail.f5.com
Address: 208.85.210.139
Name: mail.f5.com
Address: 208.85.209.139
```

Because DNS is distributed, no one controls it, which is something that works well on a global scale of interconnected disparate networks. It's also cheap and easy to run and query, with many different services available in both commercial and open-source implementations. Lastly, DNS exerts the strong inertia of being the first such system with deep legacy and dependence sunk into to the Internet's infrastructure. Because of their importance in keeping things flowing, DNS entries have timeouts that control how long entries are cached, which means that if DNS services are unavailable, clients can still run for a while off the last entry they queried. This time-to-live is server configurable and is balanced against how often you want to change your DNS entries[iv]. A long timeout means entries stay up, but changes propagate slowly.

Attacking DNS

DNS is too important to do without, but it's difficult to defend. In fact, DNS services are an excellent target for attack. Taking out an organization's DNS service renders it unreachable to the rest of the world except by IP address. If "f5.com" failed to be published online, every single Internet site and service we ran would be invisible. This means web servers, VPNs, mail services, file transfer sites—everything. Even worse, if hackers could change the DNS records, then they could redirect everyone to sites they controlled. Imagine going to "www.f5.com" and landing on a page full of banner ads. Since DNS is built upon cooperation between millions of servers and

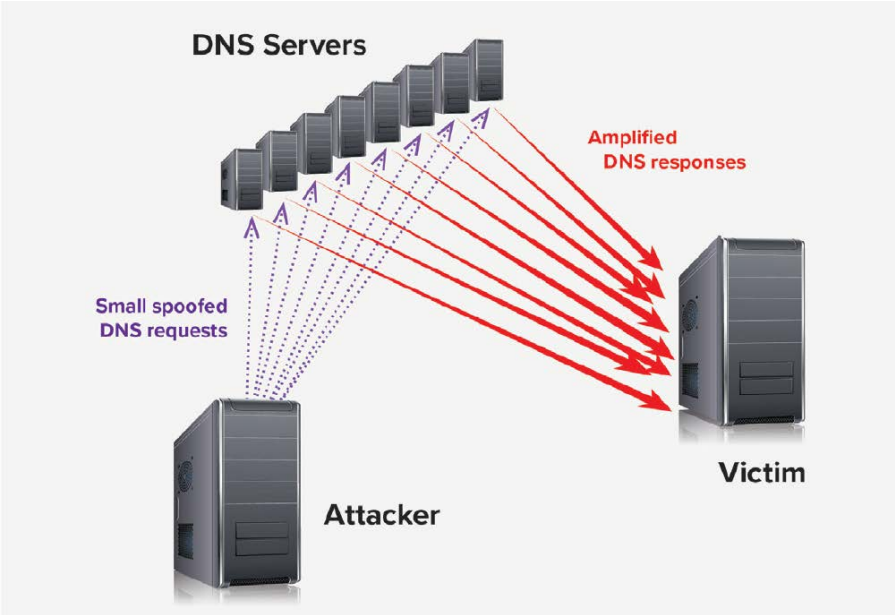
clients over insecure and unreliable protocols, it is uniquely vulnerable to disruption, subversion, and hijacking. Here's a quick rundown of the known major DNS attacks.

Denial of Service

Denial-of-service attacks are not limited to DNS, but taking out DNS decapitates an organization. Why bother flooding thousands of web sites when killing a single service does it all for you? The most famous DoS attacks against DNS are the recent Dyn, Inc. DDoS attacks which exceeded 40 gigabytes of noise blared at their DNS services. Dyn was running DNS services for many major organizations, so when they were drowned by a flood of illegitimate packets, so were companies like Amazon, Reddit, FiveThirtyEight, and Visa[v].

There are many ways to knock out DNS service, the simplest being a stream of garbage from thousands of compromised hosts (bots) in a DDoS attack. Instead of clogging up the pipe, attackers can also overwork the server with DNS Query Floodvi attacks from thousands of bots.

DNS can also be subverted for use as a denial-of-service weapon against other sites by way of DNS Amplification/Reflection[vii]. This works because DNS almost always returns a larger set of data than what was queried. A simple DNS query asking for F5.com only amounts to a few hundred bytes at most, while the response will be several orders of magnitude larger. This way an attacker can amplify network traffic through DNS servers, building up a tsunami from a ripple. Since DNS runs over UDP, it's a simple matter for attackers to craft fake packets spoofing a query source, so if they can fake thousands of queries from the victim's IP address, that tsunami of responses will return to overwhelm the victim. A bonus for the attacker is that, to the victim, it will appear as if a huge number of DNS servers are attacking it. All the while, the attacker stays safely hidden.



A DNS amplification attack floods the victim's server with a tsunami of fake requests.

DNS Hijacking

Who owns what domain name and what DNS servers are designated to answer queries are managed by Domain Registrars[viii]. These are commercial services, such as GoDaddy, eNom, and Network Solutions Inc., where there are registered accounts storing this information. If attackers can hack those accounts, they can repoint a domain to a DNS server they control. Attacks like this have affected the New York Times[ix], LinkedIn, Dell, Harvard University, Coca Cola, and many others.

DNS Server Vulnerabilities

Because DNS services are software, they are likely to contain bugs. It's possible that some of these bugs will create software vulnerabilities that attackers can exploit. That's just the way it is with all software written by imperfect carbon-based life forms. Luckily, DNS is old (so we've had time to find most of the bugs) and simple (so bugs are easy to spot), but problems have cropped up. In 2015, there was a rather significant hole found in BIND, an open-source DNS server running much of the Internet[x]. Called CVE-2015-5477[xi] (no cute name, thank you), BIND allowed an attacker to crash a DNS server with a single crafted query[xii].

Another software vulnerability in DNS servers is the Recursive DNS spoof cache poisoning[xiii] technique, which means that an attacker can temporarily change DNS database entries by issuing specifically crafted queries.

Unauthorized DNS Changes

If you've got a server, someone must manage it. That means that you are dependent on how strongly you are authenticating the admins to that server as well as ensuring the trustworthiness and competence of those admins. In practice, this vulnerability is usually realized by accident when an admin fat-fingers a DNS change or incorrectly manages the DNS servers. Because of the nature of DNS records, changes to DNS are cached by query clients, so mistakes can sometimes take hours or days to unwind across the Internet.

DNS Data Leakage

You can't run an unauthenticated Internet database full of important information without the occasional risk of leaking out something important. Attackers will often repeatedly query DNS servers as a prelude to an attack, looking for interesting Internet services that may not be widely known. For example, an organization may have a site called vpn.example.com which it doesn't advertise to anyone except its employees. If an attacker discovers this site, they've just found a new potential target in an attack. DNS records can also aid phishing expeditions by using known server names in their phony baloney emails.

Many organizations run DNS on the inside of the network, advertising local area network (LAN) resources for workstations. Some smaller organizations run split-horizon DNS servers[xiv] that offer up Internet DNS services to the world as well as these LAN-based DNS services on the same box. A wrong configuration on that DNS server can lead to some devastating DNS data leakages as internal names and addresses are shared with attackers. Even giants can be tripped up by this seemingly simple vulnerability.[xv]

DNS Man-in-the-Middle

Once again, the easily spoofed protocol UDP that DNS uses is the weak link. In this case, an attacker inline between the victim and the DNS server they're querying can intercept and monkey with DNS queries. It's a pretty easy attack to pull off if you're on the same wire or wireless as the victim or DNS server. What can you do with this? Well, an F5 researcher found a way to use it to steal Microsoft Outlook credentials[xvi]. So, it's an attack that shouldn't be taken lightly.

A good defense against this to run DNS Security Extensions (DNSSEC) on the DNS server, which adds public-privacy cryptographic keys to authenticate records. Adoption is slowly growing[xvii] but DNSSEC can be hard to implement.

Defending DNS

Given these attacks, how do you defend such a vital service? Many organizations simply choose to outsource their DNS services, which moves the responsibility to someone who should have more resources and skills to defend it. Not all DNS vulnerabilities are reduced this way, but many are. Here's a breakdown of the DNS attacks, some threat potentials, and recommended defenses:

Attack Type	Threat Likelihood	Impact Potential	Possible Mitigations
Denial-of-service	**High**. Attacks are common and easy to pull off, for example, by renting a botnet.	**High**. Loss of all Internet service during the attack.	• Distribute multiple DNS servers. • Harden DNS (See NIST Secure DNS Deployment Guide). • Outsource DNS services to an organization equipped to deal with DOS.
DNS query flood	**Low**. Attacks are uncommon as of now.	**High**. Loss of all Internet service during the attack.	• Harden DNS (See NIST Secure DNS Deployment Guide). • Outsource DNS services to an organization equipped to deal with DOS.
DNS amplification/ reflection	**Medium**. Attacks are common but affect the Internet community as a whole.	**Low** to DNS server owner, but higher to victim.	• Harden DNS (See NIST Secure DNS Deployment Guide).
DNS hijacking	**Medium**. Attacks are not common but do happen with some regularity. Many registrars now offer better authentication mechanisms and verification.	**High**. Complete loss of control over integrity of DNS records. In the best case, this means outages. In the worst case, your domains resolve to sites hosting insults, fraud, porn, spam, or worse.	• Ensure domain registrar accounts use strong authentication. • Monitor domain registrar accounts to ensure information is kept up to date.
DNS vulnerabilities	**Low.** Vulnerabilities in DNS services are uncommon but not unknown.	**High.** Can lead to outages, DNS record hijacking, or even breach of internal networks.	• Ensure DNS servers are patched in a timely fashion. • Outsource DNS services to an organization equipped to deal with DOS.
Unauthorized DNS changes	**Medium**. One of the most common causes of DNS outages is admin mistakes.	**High**. Can lead to outages that can linger for days.	• Least privilege to limit access to DNS servers. • Strong change control and review of all changes. • Outsource DNS services to an organization equipped to deal with DOS.
DNS data leakage	**Medium.** A very common method for attackers to do reconnaissance.	**Low.** This information usually needs to be combined with another attack method to be useful.	• Least privilege to limit access to DNS servers. • Strong change control and review of all changes. • Review DNS records on a regular basis.
DNS man-in-the-middle	**Low**. Not common but can happen.	**Low**. Usually limited to a single victim or group of victims. For that victim, consequences can be high.	• Implement DNSSEC. • Outsource DNS services to an organization that includes DNSSEC.

Hardening DNS

The National Institute of Standards (NIST) has published the *Secure Domain Name System (DNS) Deployment Guide*[xviii], which is a comprehensive document on securing DNS. This is to be used in

A good way to gauge the strength of your DNS services is to engage a penetration test with DNS in scope to check. It's not foolproof, but it can give you an idea of what you might have missed.

Conclusion

It's obvious that DNS is a critical piece of Internet infrastructure. As security guru Dan Geer says, "Risk is a consequence of dependence."[xix] We are stuck with DNS, so better make sure it's reliable and incorruptible. The future of the Internet depends on it.

[i] https://www.reddit.com/r/ProgrammerDadJokes/comments/50lqcx/i_would_tell_you_a_udp_joke_but_you_probably/
[ii] https://tools.ietf.org/html/rfc882
[iii] https://en.wikipedia.org/wiki/BIND
[iv] https://www.dnsknowledge.com/whatis/time-to-live-ttl/
[v] https://en.wikipedia.org/wiki/2016_Dyn_cyberattack
[vi] https://f5.com/glossary/dns-flood-query-flood
[vii] https://deepthought.isc.org/article/AA-00897/0/What-is-a-DNS-Amplification-Attack.html
[viii] http://www.domainstate.com/registrar-stats.html
[ix] http://www.pcworld.com/article/2047628/spear-phishing-led-to-dns-attack-against-the-new-york-times-others.html
[x] https://www.isc.org/downloads/bind/
[xi] https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-5477
[xii] https://www.exploit-db.com/exploits/37721/
[xiii] http://www.networkworld.com/article/2277316/tech-primers/tech-primers-how-dns-cache-poisoning-works.html
[xiv] https://en.wikipedia.org/wiki/Split-horizon_DNS
[xv] https://www.rcesecurity.com/2017/03/ok-google-give-me-all-your-internal-dns-information/
[xvi] https://f5.com/labs/articles/threat-intelligence/identity-threats/how-three-low-risk-vulnerabilities-become-one-high-24995
[xvii] http://www.internetsociety.org/deploy360/dnssec/statistics/
[xviii] http://dx.doi.org/10.6028/NIST.SP.800-81-2
[xix] http://queue.acm.org/detail.cfm?id=2479677