Smayan Daruka
CSEC-380
Professor Chaim Sanders

DNS & DNSSEC vulnerabilities

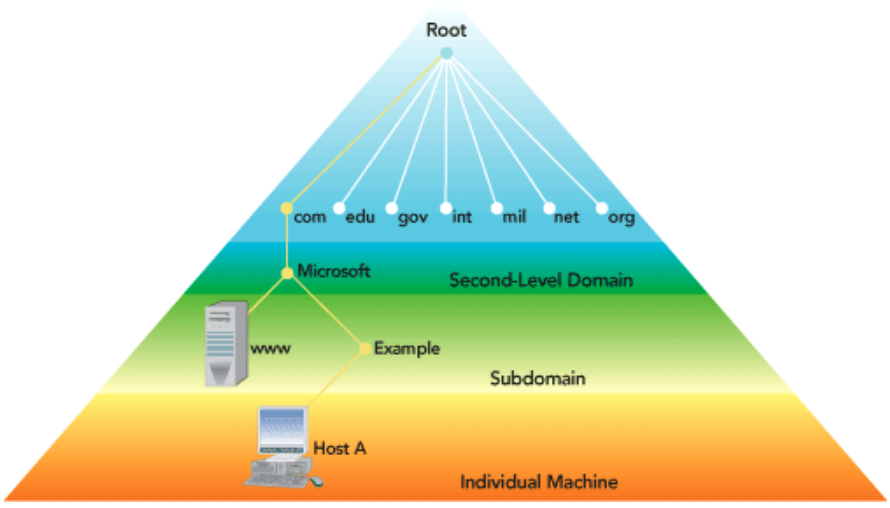
Introduction

Domain Name System (DNS) is commonly referred to as the phonebook of the Internet. It is an absolutely essential service that provides a hierarchical and decentralized naming system designed in the mid-1980s. Security of the protocol was not a major design consideration back then. Ever since the Internet became available for public use, security measures have been implemented to make the protocol secure. This blog post talks about the following:

- DNS
    - DNS record types
    - DNS Vulnerabilities & Attacks
- DNSSEC
    - DNSSEC record types
- NSEC vulnerability
- Mitigation steps
- Conclusion

DNS

DNS stands for Domain Name System and it is a hierarchical naming system that translates domain names into IP addresses. DNS is the service that allows users to type in names (words) instead of numbers when trying to visit a website. The image below is fairly helpful in understanding how DNS is implemented. DNS is distributed, in that, its information is spread among thousands of servers all over the world. Names are processed from right to left. At the top, there are 13 root name servers, and they contain all the information for the generic top-level domains (TLDs) followed by each TLD containing information for the subdomains and so forth. As can be seen from the image, the hierarchy becomes more specific as we move downwards.



Smayan Daruka
CSEC-380
Professor Chaim Sanders

DNS resolution allows us to convert a hostname to an IP address. For instance, if a user wanted to visit www.google.com, the host first looks in its cache to see if it already has an IP address for that hostname. If it does not, it sends a DNS request to its DNS server. At this point, the DNS server checks its cache as well and if it is not able to find a match, it sends a DNS request to one of the root servers. The root servers look in their zone files and if they are not able to find a match either, the root server refers the querying DNS server (referring to the host in our case) to the TLD that the domain is in. In our case, the TLD is ".com". The DNS server then queries the ".com" TLD for the IP address of www.google.com. If the IP address is not available in cache, the querying DNS server is then referred to the authoritative DNS server for the domain "google.com". Lastly, the IP address is sent back to the querying DNS server, which caches it for future use, and also sends it to the host.

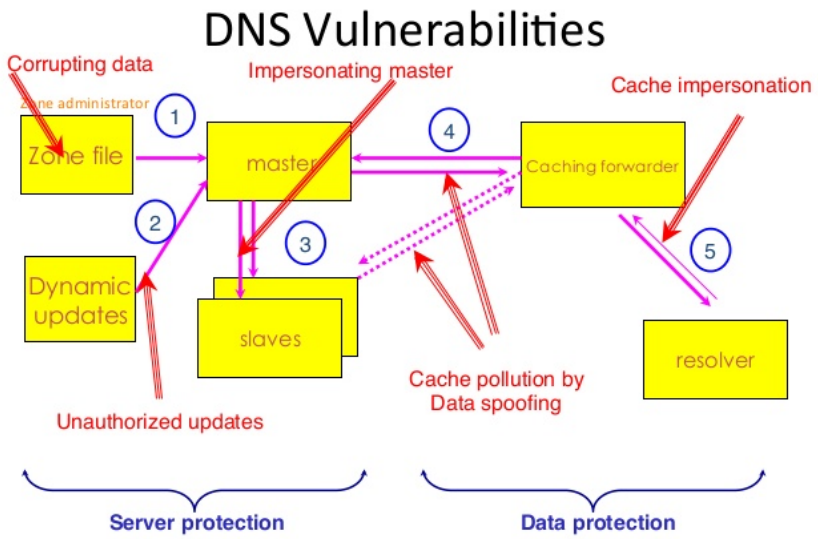
DNS Record Types

DNS records reside in the authoritative name servers and provide a variety of information. There are a lot of record types but the most common ones and their descriptions are listed below:

- A record – has the IP address of the domain
- CNAME record – refers to an alias of a domain
- MX record – provides the mail server information for the domain
- NS record – provides the name of the authoritative name server for the domain
- PTR record – provides a domain name for an IP address (opposite of the A record)
- SOA record – provides general info about the domain

DNS Vulnerabilities & Attacks

The diagram below shows a few places in the whole DNS cycle where things can go wrong.



For instance, an attacker could corrupt the zone file and do unauthorized updates (1 and 2). An attacker could also do DNS hijacking and impersonate the master (3) or do DNS cache poisoning and impersonation (4 and 5).

Smayan Daruka
CSEC-380
Professor Chaim Sanders

The top 5 DNS attacks are as follows:

- DNS amplification – it is a popular form of asymmetrical distributed denial of service (DDoS) attack which relies on the use of publically available open DNS servers to overload a target DNS server. The technique consists of an attacker sending a DNS lookup request to a public DNS server with the spoofed source IP address of the target DNS server. So when the DNS server responds, the answer gets sent to the target DNS server. To maximize the amplification effect, attackers usually request as much zone information as possible. The reason for this is that most of the times, DNS responses are much larger than DNS requests so the attack is further amplified, thus the name.
- DNS cache poisoning – this type of attack diverts traffic from legitimate servers to fake servers. It is sometimes also referred to as DNS spoofing. Corrupt DNS data is entered into the DNS resolver's cache which causes the server to return an incorrect IP address. DNS cache poisoning can spread pretty fast if ISPs (Internet Service Provider) receive the corrupt DNS entries and pass it onto personal devices which then store it in the cache until the TTL (time-to-live) expires. Once a computer is poisoned, the user will be visiting a fake website that is spoofed to look like the legitimate website.
- DNS hijacking – this is a type of attack where DNS queries get redirected to fake servers. The most common way to achieve this is by overriding the DNS settings of a host and pointing it to the fake DNS server which is controlled by the attacker.
- DNS flood – it is a popular form of symmetrical DDoS attack and it attempts to exhaust server-side resources with a large number (flood) of UDP requests. As there is no established connection, spoofing is a lot easier in this scenario. An attacker uses multiple machines or a botnet to send DNS requests with spoofed source IP addresses. One subset of a DNS flood attack is a DNS NXDOMAIN (non-existent) flood attack where the attacker sends queries for records that are non-existent so as to fill up the server's cache with useless data.
- DNS tunneling – this is a type of attack where an attacker encodes data from other programs or protocols in DNS queries and responses. This type of attack requires access to an internal DNS server. Simply put, as long as someone can do domain name lookups, one can do DNS tunneling to access blocked websites for instance.

DNSSEC

Domain Name System Security Extensions (DNSSEC) are, as the name suggests, a set of extensions to DNS that provide a way to authenticate the origin of DNS records and verify the integrity of data. The whole idea behind DNSSEC is to add security to DNS while also ensuring that it is backwards compatible. It is a protocol modification for DNS queries and responses, and it depends on public-key encryption. DNSSEC allows a server to sign replies and the client to validate said signature.

The primary purpose of DNSSEC is to make sure that users aren't redirected to fraudulent IP addresses. DNSSEC gives the ability to protect against man-in-the-middle attacks and cache poisoning by matching cryptographic signatures to DNS records. Due to the protocol modification, DNSSEC requires clients to have DNSSEC-capable resolvers. DNSSEC uses a rigid trust model and the chain of trust flows from the parent zone to the child zone. The parent zone essentially vouches for the child zone and the process repeats.

Smayan Daruka
CSEC-380
Professor Chaim Sanders

DNSSEC Record Types

DNSSEC creates a secure domain name system by adding cryptographic features to existing DNS records. To facilitate with signature validation, the following record types are added:

- RRSIG – Resource Record Signature (cryptographic signature) – it is used to sign the resource record set which refers to all the records of a particular type. It is essentially a digital signature and each RRSIG is associated with a DNSKEY.
- DNSKEY – it is a cryptographic public key and there are two types:
    - KSK – Key Signing Key – refers to the key that is used to sign DNSKEY records
    - ZSK – Zone Signing Key – refers to the key that is used to sign all other records in the domain
- DS – Delegation Signer – it is the hash of DNSKEY.
- NSEC – Next Secure Record – lists the previous and following record in case of a non-existent domain
- NSEC3 – Same as NSEC but it is the hashed version

The DS is always stored one level above the current domain. For instance, the DS record for the ".com" TLD will be stored in the root zone. NSEC and NSEC3 records provide denials-of-existence records. The table below shows where each record goes in the chain and how they are related:

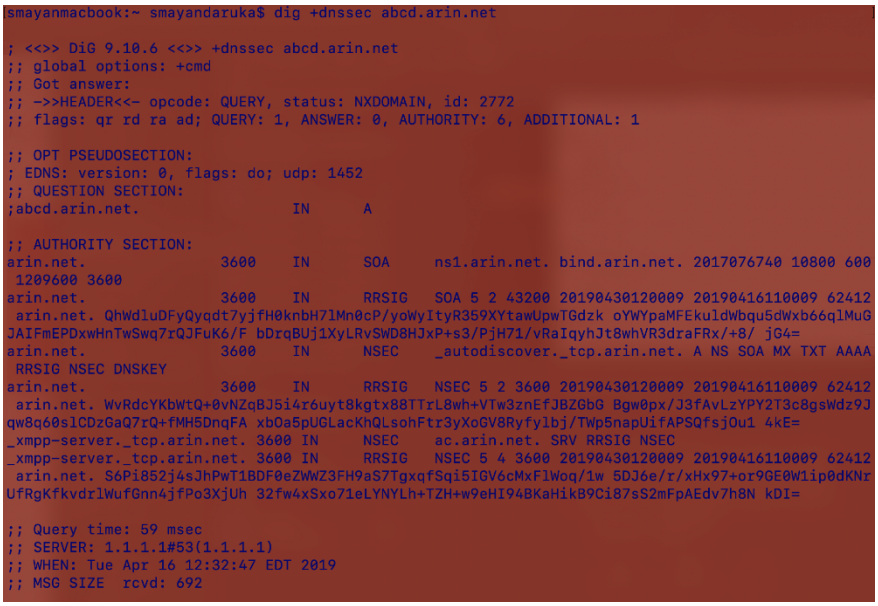
Owner	**Record Type**	**Zone**
www.google.com.	A	google.com.
www.google.com.	RRSIG	google.com.
google.com.	DNSKEY	google.com.
google.com.	DS	com.
google.com.	RRSIG	com.
com.	DNSKEY	com.
com.	DS	.
com.	RRSIG	.
.	DNSKEY	.

As can be seen in the table above, every record type is validated and protected by the record type in the following row. In this example, the "google.com" domain provides a RRSIG of the A record in the first row. The same domain also provides a DNSKEY to validate the "google.com" domain itself. Then the ".com" TLD provides the DS record for the owner on the left. The ".com" TLD also provides a RRSIG and DNSKEY for "google.com" and ".com" respectively. Finally, the root server provides a DS and RRSIG to validate the ".com" domain and it provides its own DNSKEY. At the very end, there is no DS record to validate the root zone, and this is where we have to trust individuals that partake in a Root Signing Ceremony which is held in a public and audited manner.

Smayan Daruka
CSEC-380
Professor Chaim Sanders

NSEC Vulnerability

The NSEC vulnerability is one instance to show how DNSSEC was implemented incorrectly for a domain and how we can map out a complete domain.



```
smayanmacbook:~ smayandaruka$ dig +dnssec abcd.arin.net

; <<>> DiG 9.10.6 <<>> +dnssec abcd.arin.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 2772
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1452
;; QUESTION SECTION:
;abcd.arin.net.                 IN      A

;; AUTHORITY SECTION:
arin.net.               3600    IN      SOA     ns1.arin.net. bind.arin.net. 2017076740 10800 600
 1209600 3600
arin.net.               3600    IN      RRSIG   SOA 5 2 43200 20190430120009 20190416110009 62412
 arin.net. QhWdluDFyQyqdt7yjfH0knbH7lMn0cP/yoWyItyR359XYtawUpwTGdzk oYWYpaMFEkuldWbqu5dWxb66qlMuG
JAIFmEPDxwHnTwSwq7rQJFuK6/F bDrqBUj1XyLRvSWD8HJxP+s3/PjH71/vRaIqyhJt8whVR3draFRx/+8/ jG4=
arin.net.               3600    IN      NSEC    _autodiscover._tcp.arin.net. A NS SOA MX TXT AAAA
 RRSIG NSEC DNSKEY
arin.net.               3600    IN      RRSIG   NSEC 5 2 3600 20190430120009 20190416110009 62412
 arin.net. WvRdcYKbWtQ+0vNZqBJ5i4r6uyt8kgtx88TTrL8wh+VTw3znEfJBZGbG Bgw0px/J3fAvLzYPY2T3c8gsWdz9J
qw8q60slCDzGaQ7rQ+fMH5DnqFA xbOa5pUGLacKhQLsohFtr3yXoGV8Ryfylbj/TWp5napUifAPSQfsjOu1 4kE=
_xmpp-server._tcp.arin.net. 3600 IN     NSEC    ac.arin.net. SRV RRSIG NSEC
_xmpp-server._tcp.arin.net. 3600 IN     RRSIG   NSEC 5 4 3600 20190430120009 20190416110009 62412
 arin.net. S6Pi852j4sJhPwT1BDF0eZWWZ3FH9aS7TgxqfSqi5IGV6cMxFlWoq/1w 5DJ6e/r/xHx97+or9GE0W1ip0dKNr
UfRgKfkvdrlWufGnn4jfPo3XjUh 32fw4xSxo71eLYNYLh+TZH+w9eHI94BKaHikB9Ci87sS2mFpAEdv7h8N kDI=

;; Query time: 59 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Tue Apr 16 12:32:47 EDT 2019
;; MSG SIZE  rcvd: 692
```

As can be seen in the image above, if we query for a non-existent domain (abcd.arin.net), we get the NSEC records that list the previous and next record. In this case, the previous record is the "_autodiscover._tcp.arin.net" one and the latter is "ac.arin.net".

```
smayanmacbook:~ smayandaruka$ dig +dnssec test2.arin.net

; <<>> DiG 9.10.6 <<>> +dnssec test2.arin.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 1138
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1452
;; QUESTION SECTION:
;test2.arin.net.                        IN      A

;; AUTHORITY SECTION:
arin.net.               3600    IN      SOA     ns1.arin.net. bind.arin.net. 2017076740 10800 600
 1209600 3600
arin.net.               3600    IN      RRSIG   SOA 5 2 43200 20190430120009 20190416110009 62412
 arin.net. QhWdluDFyQyqdt7yjfH0knbH7lMn0cP/yoWyItyR359XYtawUpwTGdzk oYWYpaMFEkuldWbqu5dWxb66qlMuG
JAIFmEPDxwHnTwSwq7rQJFuK6/F bDrqBUj1XyLRvSWD8HJxP+s3/PjH71/vRaIqyhJt8whVR3draFRx/+8/ jG4=
arin.net.               3600    IN      NSEC    _autodiscover._tcp.arin.net. A NS SOA MX TXT AAAA
 RRSIG NSEC DNSKEY
arin.net.               3600    IN      RRSIG   NSEC 5 2 3600 20190430120009 20190416110009 62412
 arin.net. WvRdcYKbWtQ+0vNZqBJ5i4r6uyt8kgtx88TTrL8wh+VTw3znEfJBZGbG Bgw0px/J3fAvLzYPY2T3c8gsWdz9J
qw8q60slCDzGaQ7rQ+fMH5DnqFA xbOa5pUGLacKhQLsohFtr3yXoGV8Ryfylbj/TWp5napUifAPSQfsjOu1 4kE=
t.arin.net.             3600    IN      NSEC    tinnie.arin.net. A AAAA RRSIG NSEC
t.arin.net.             3600    IN      RRSIG   NSEC 5 3 3600 20190430120009 20190416110009 62412
 arin.net. BtwybWxsSsQV+zcGZ38qGNDLdPEEb85NVfcuK6kyi5LUn7ubstEAlsWQ RBCkNuVg8WXhp7UUHbI3ifscuUUVA
bTNRdC1JFoYgfB1yZD6XuY4qoTB +37cK/1lFOVIVMIQ94456Ye4//gjwUUEWvTr7nyf5kuoco3gaF+qCcx/ qdY=

;; Query time: 47 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Tue Apr 16 12:50:10 EDT 2019
;; MSG SIZE  rcvd: 681
```

As can be seen in the image above, if we query for a non-existent domain (test2.arin.net), we get the NSEC records that tell us the next record is "tinnie.arin.net".

Following this method allows us to map out an entire domain. This method is a little time consuming since it is a manual process of querying over and over.

Smayan Daruka
CSEC-380
Professor Chaim Sanders

Mitigation Steps

Since NSEC is the plaintext version of the record, the easiest way to mitigate the vulnerability above is to use NSEC3 since it hashes the record. It essentially prevents zone enumeration and uses different algorithms to hash the record, most notable SHA-1. An important thing to note is that this method does not make it fool-proof to an attack. Another implementation is to provide the NSEC3 record of the hash of the next domain. Although this cannot be easily implemented since it requires real-time computation.

Conclusion

Domain Name System Security Extensions (DNSSEC) provide protection against DNS spoofing, DNS cache poisoning, man-in-the-middle attacks, and more. They also increase trust for normal online activities. With all the advantages, there are a couple of disadvantages as well. Implementing DNSSEC adds a lot more complexity to both clients and servers. There is limited support from TLDs and DNS servers in a broad spectrum. While adding a layer of security, DNSSEC like any other technology, also introduces new vulnerabilities and attack vectors like the NSEC vulnerability. Overall, DNSSEC is a valuable tool that will go a long way in helping us improve trust and integrity of DNS. Adoption should grow at a fast rate as security concerns continue to exist.

Smayan Daruka
CSEC-380
Professor Chaim Sanders

REFERENCES:

IMAGES:

https://i-technet.sec.s-msft.com/dynimg/IC319093.gif

https://image.slidesharecdn.com/dnssec-cw1410840227-140916002907-phpapp01/95/dnssec-tutorial-by-champika-wijayatunga-apnic-38-6-638.jpg?cb=1410993075

WEBSITES:

https://webhostinggeeks.com/guides/dns/

https://www.cloudflare.com/learning/dns/dns-records/

https://www.us-cert.gov/ncas/alerts/TA13-088A

https://www.howtogeek.com/161808/htg-explains-what-is-dns-cache-poisoning/

https://www.cactusvpn.com/beginners-guide-online-security/dns-hijacking/

https://www.imperva.com/learn/application-security/dns-flood/

https://resources.infosecinstitute.com/dns-tunnelling/

https://usa.kaspersky.com/resource-center/definitions/dns

https://blog.cloudflare.com/dnssec-an-introduction/

https://blog.cloudflare.com/dnssec-complexities-and-considerations/