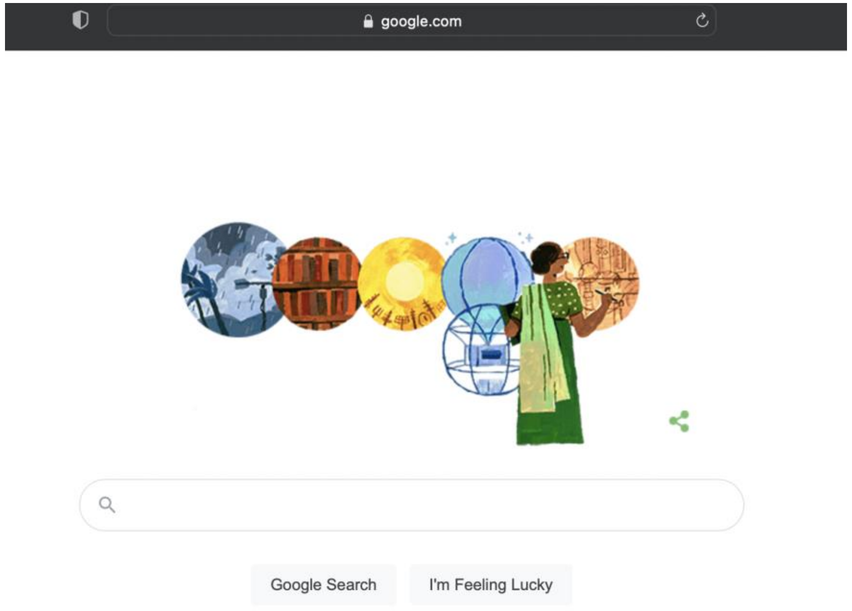
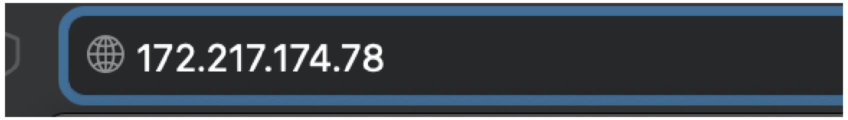
Introduction to DNS

DNS(Domain Name System)

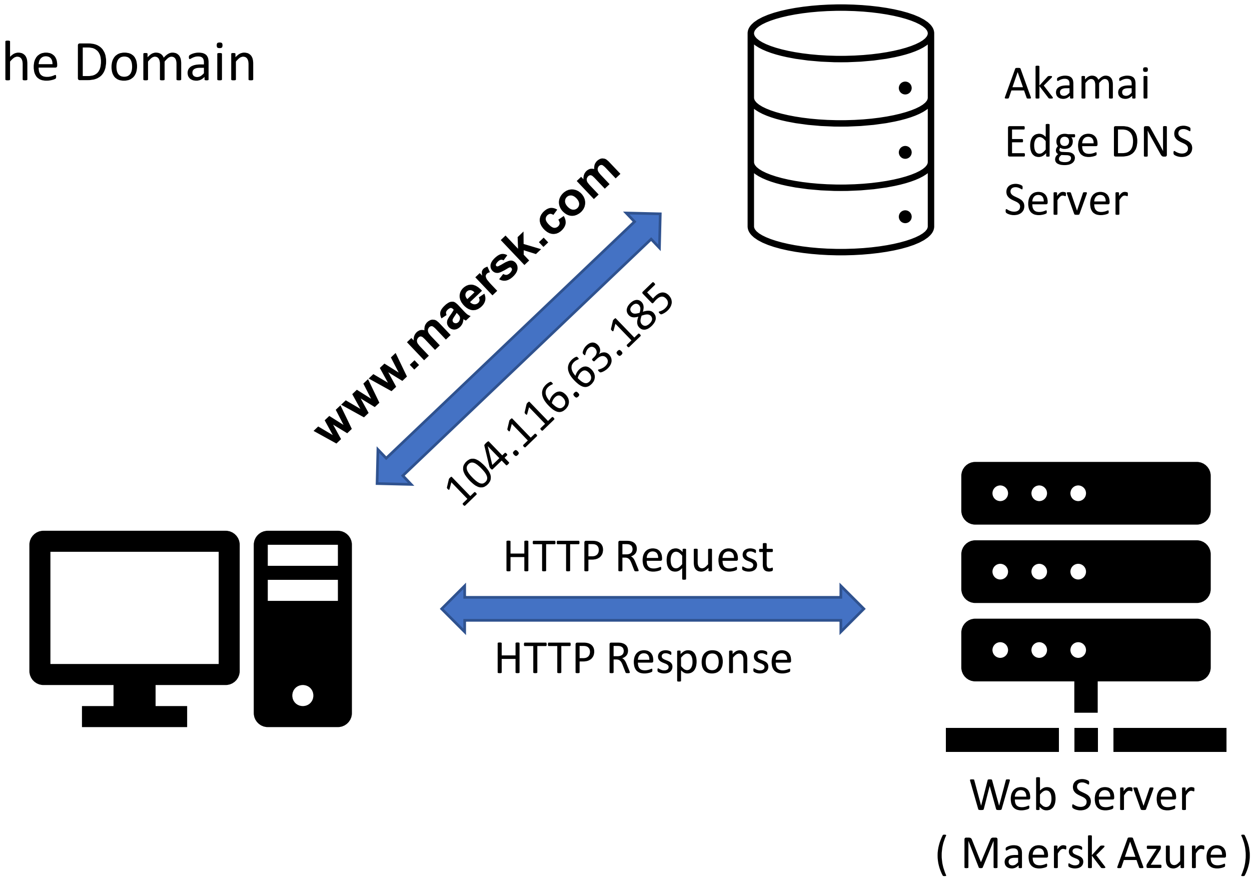
- Good Luck Remembering 172.217.174.78
- That is google.com btw
- Devices across the Internet can be identified easily


google.com


172.217.174.78

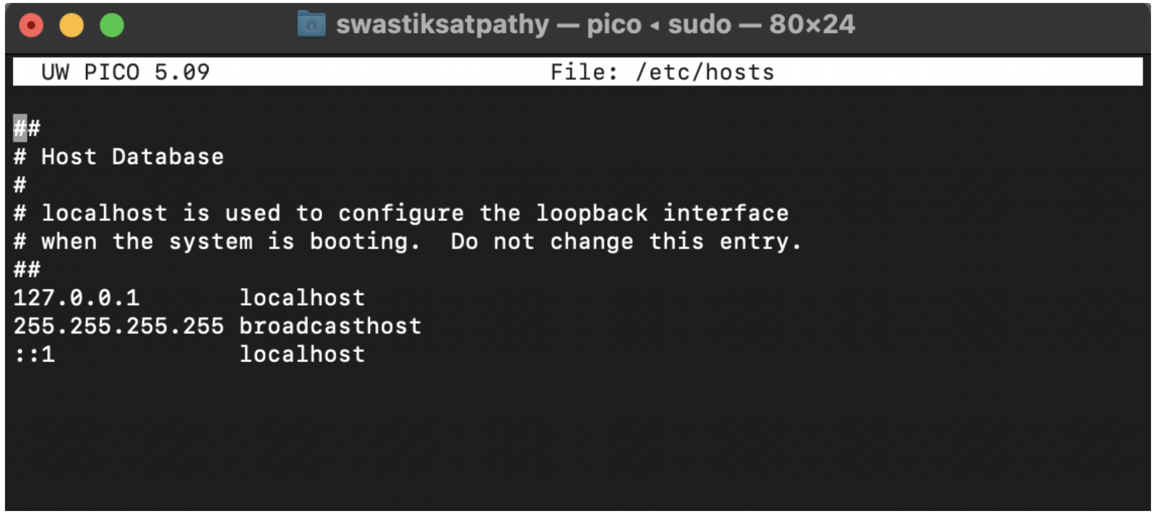
How DNS works?

- A request is made by client to their DNS server
- The DNS servers returns the IP Address of the Domain
- The client then establishes Connection with the IP returned.



History of DNS

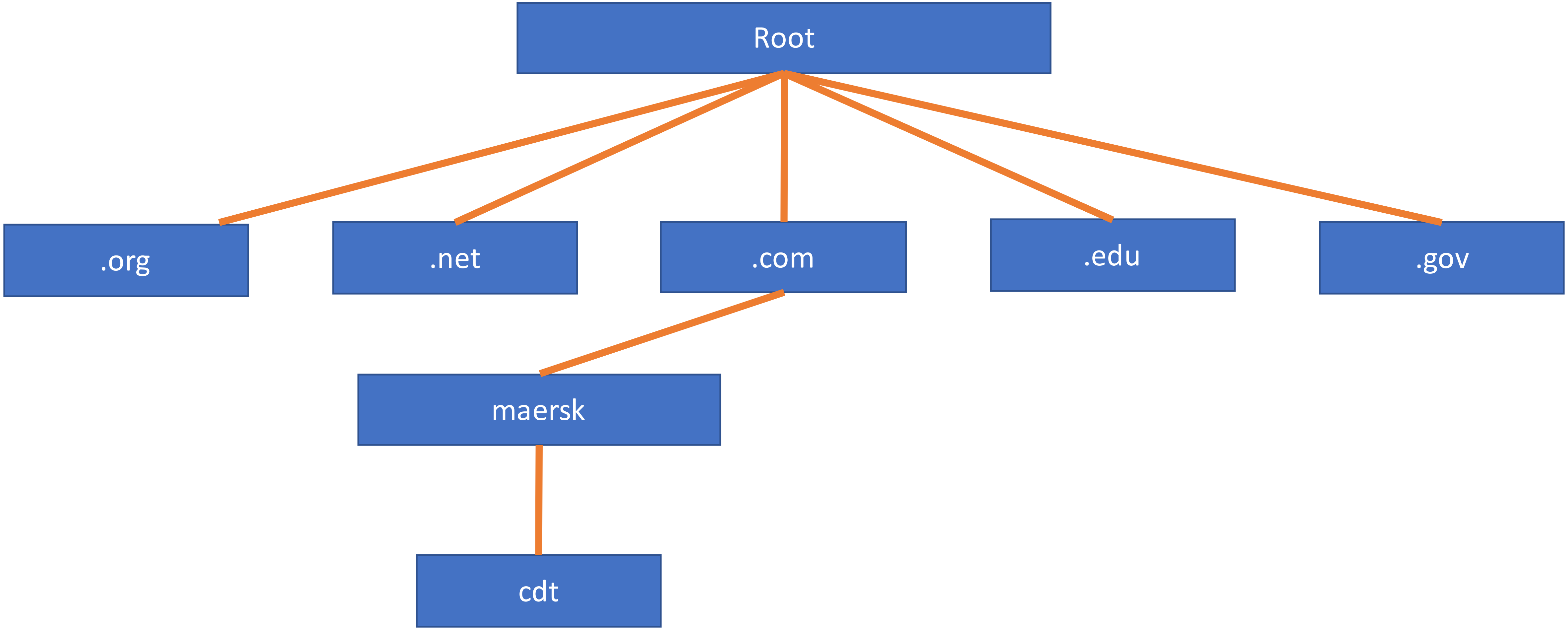
- DNS Records used to be store on Local Machines
- hosts.txt file was used to map websites
- This method consumed a lot of bandwidth
- It would take ages to onboard a new domain



```
  UW PICO 5.09                                      File: /etc/hosts

##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting.  Do not change this entry.
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
```

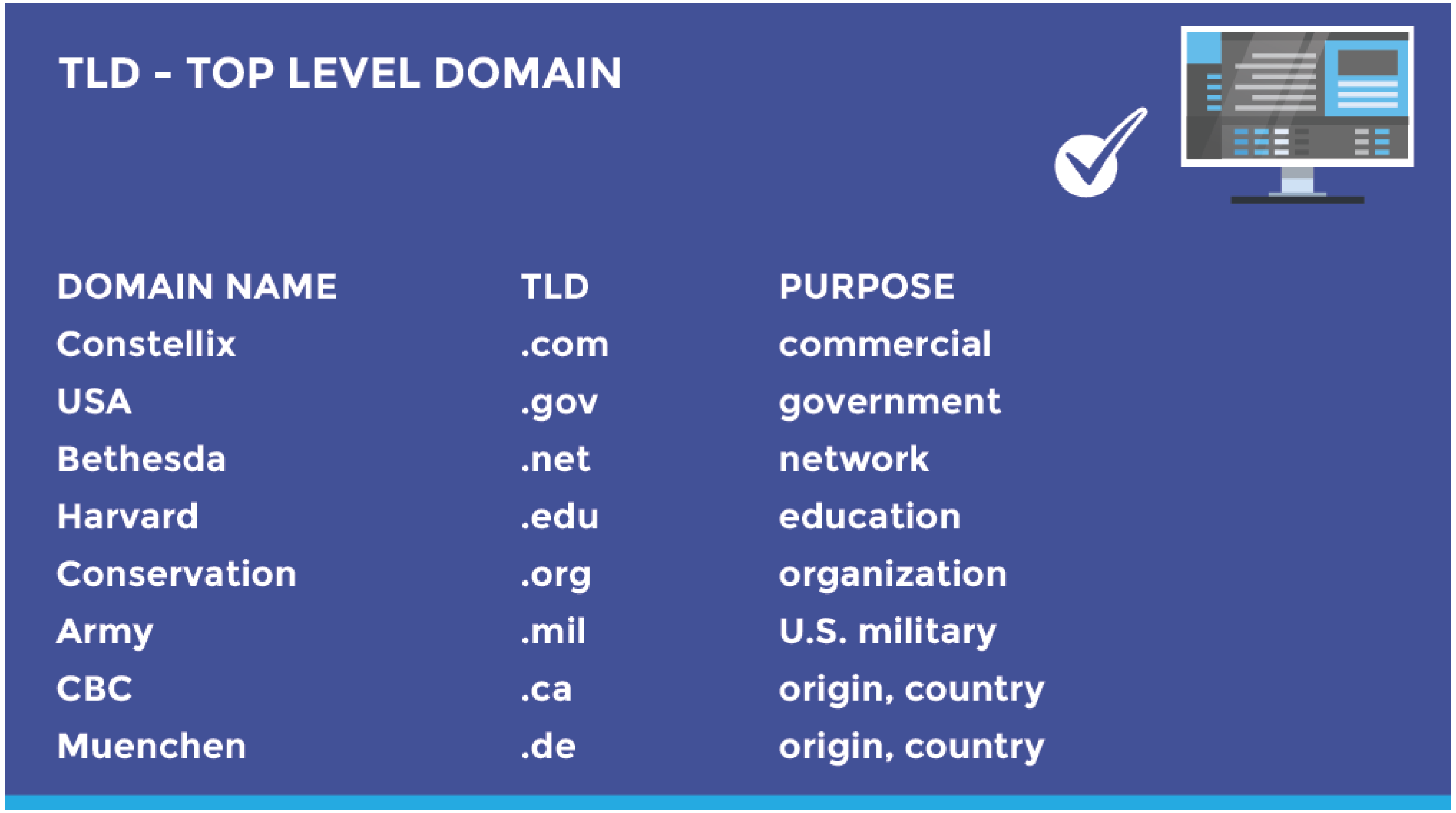
DNS Hierarchy



Classification: Internal

Top Level Domains

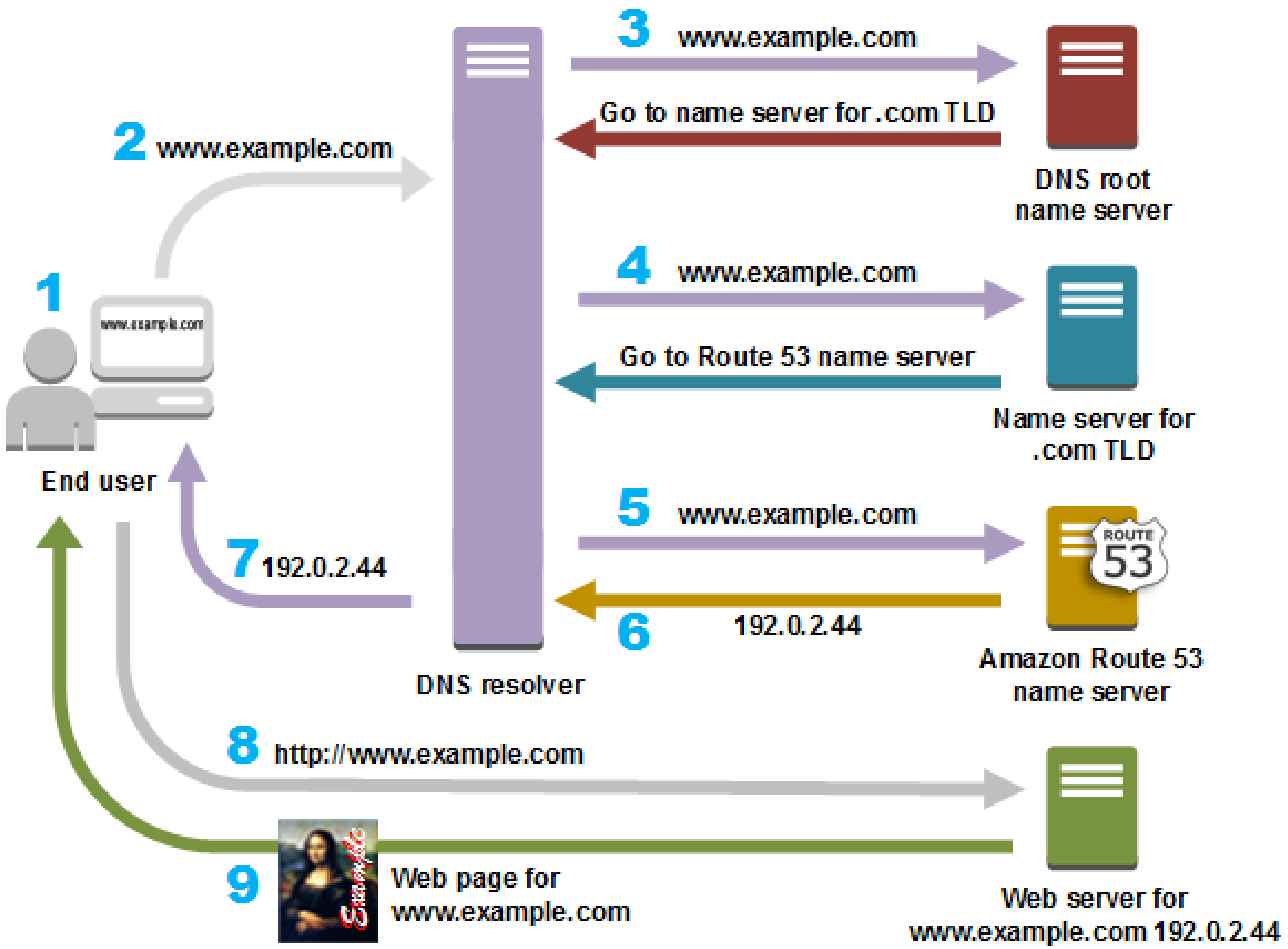
- Examples of TLDs



TLD - TOP LEVEL DOMAIN

DOMAIN NAME	TLD	PURPOSE
Constellix	.com	commercial
USA	.gov	government
Bethesda	.net	network
Harvard	.edu	education
Conservation	.org	organization
Army	.mil	U.S. military
CBC	.ca	origin, country
Muenchen	.de	origin, country

DNS Resolution

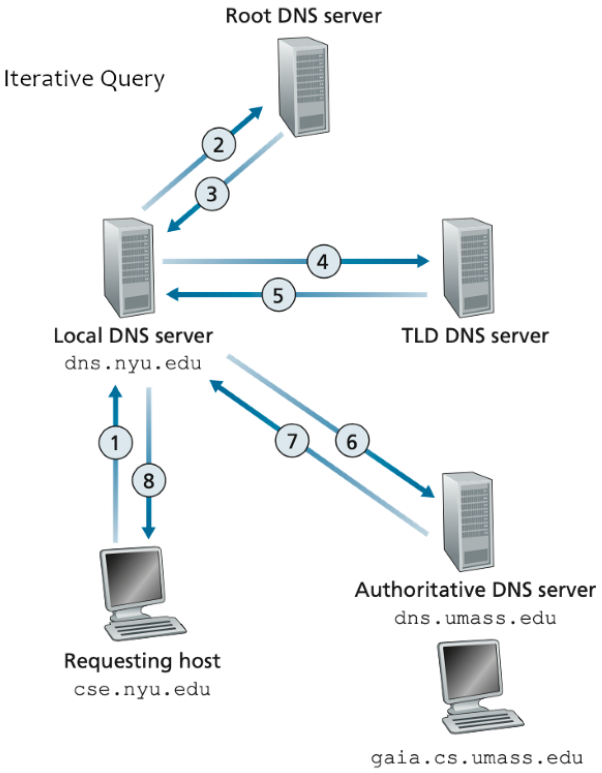


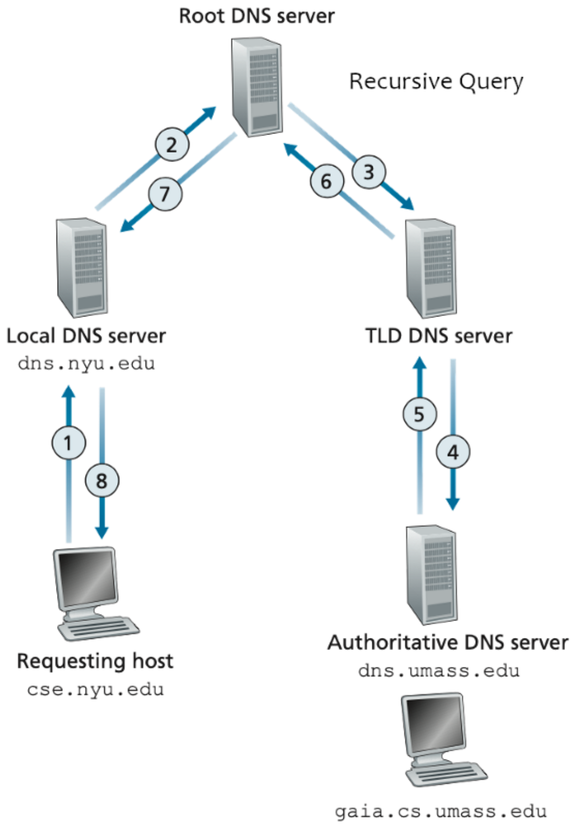
Types of Nameservers

- DNS Recursive Resolvers
- Root nameservers
- TLD nameservers
- Authoritative nameservers

DNS Query types

- Recursive query
- Iterative query
- Non-recursive query





DNS Records

- A (IPV4 Records)
- AAAA (IPV6 Records)
- CNAME (Alias)
- MX (Mail exchange records)
- NS (Nameserver records)
- PTR (Opposite of A records provides IP address instead of domain name)
- SRV (VOIP records)
- SOA (Start of Authority records)

A record

- Matching a Domain Name like (www.maersk.com) to an IPV4 address
- @ - indicates this is a record for the root domain
- 14400 value is TTL (Time to live in seconds)

www.maersk.com	record type	value	ttl
@	A	192.168.2.1	14400

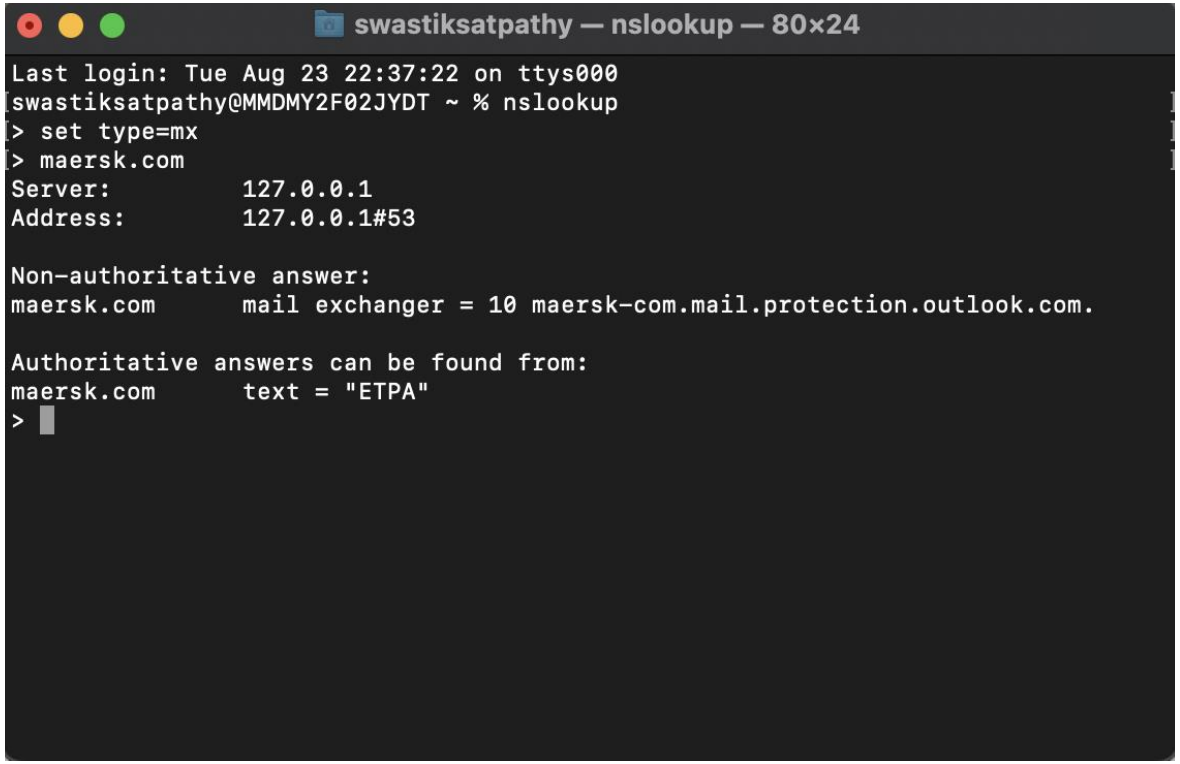
AAAA record

- This contains the IPv6 address for the domain

example.com	record type	value	TTL
@	AAAA	2606:2800:220:1:248:1893:25c8:1946	14400

MX Records

A MX record points mails to a mail server.



```
Last login: Tue Aug 23 22:37:22 on ttys000
swastiksatpathy@MMDMY2F02JYDT ~ % nslookup
> set type=mx
> maersk.com
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
maersk.com      mail exchanger = 10 maersk-com.mail.protection.outlook.com.

Authoritative answers can be found from:
maersk.com      text = "ETPA"
> 
```

example.com	record type	priority	value	TTL
@	MX	10	mailhost1.example.com	45000

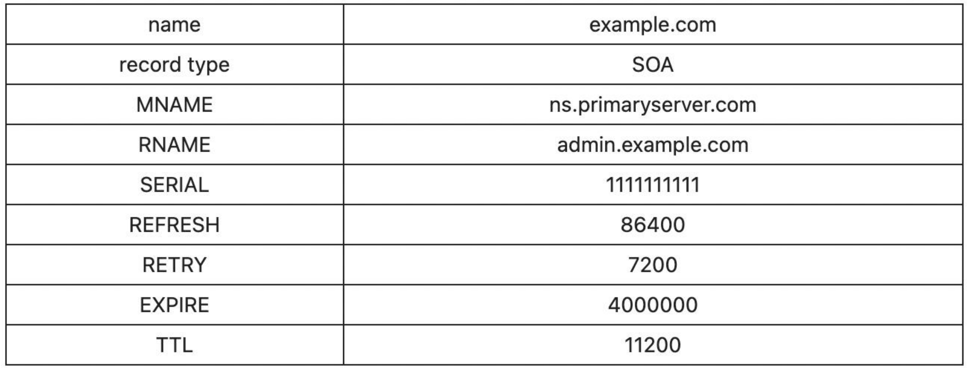
NS Records

- Stores records of nameservers

example.com	record type:	value:	TTL
@	NS	ns1.exampleserver.com	21600

SOA Record

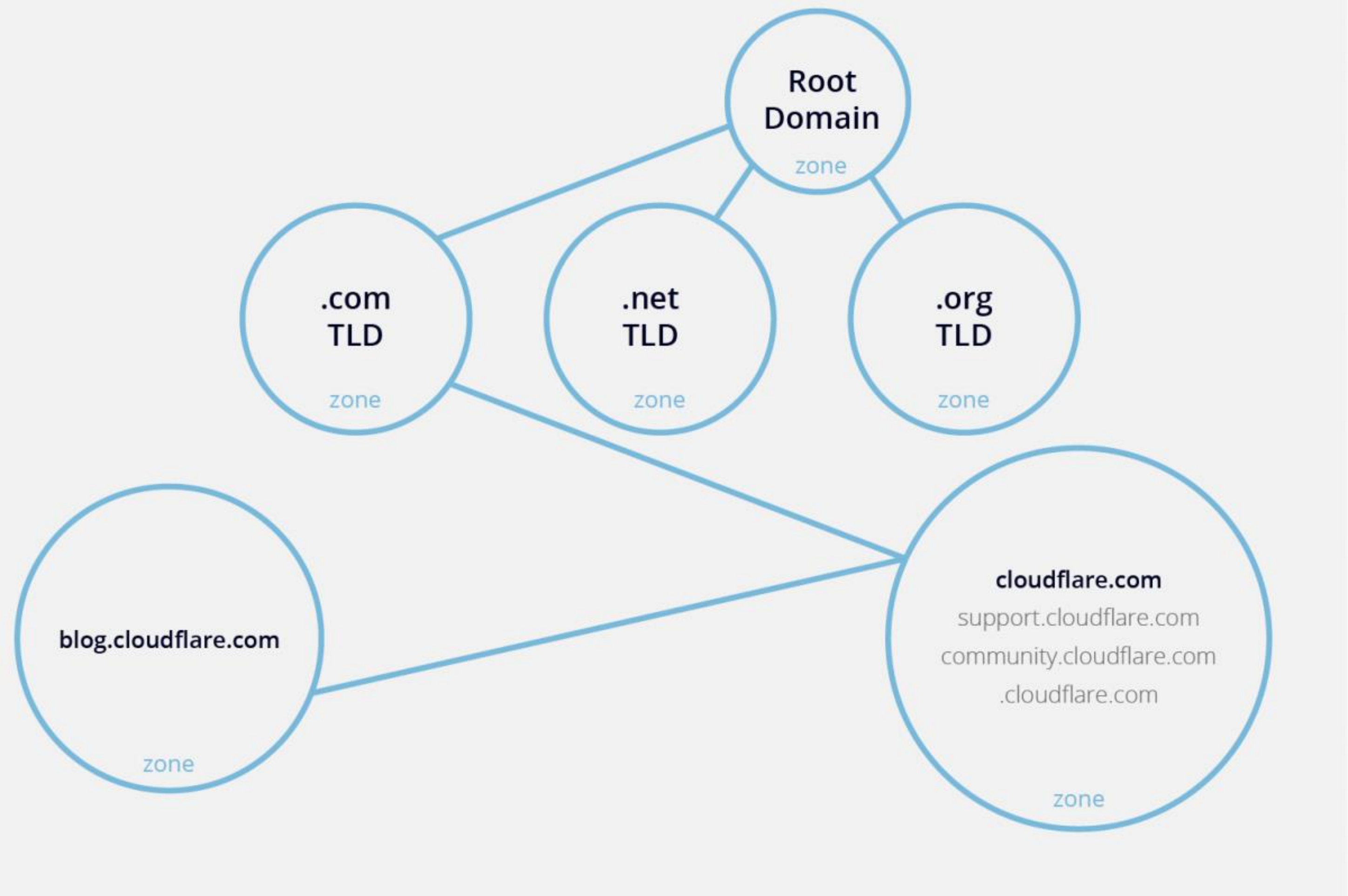
- Zone files must always start with a Start of Authority (SOA) record, which contains important information including contact information for the zone administrator.
- Contains Information about Primary and Secondary Servers
- Serial numbers can be used to apply updates across zones using Zone transfer



name	example.com
record type	SOA
MNAME	ns.primaryserver.com
RNAME	admin.example.com
SERIAL	1111111111
REFRESH	86400
RETRY	7200
EXPIRE	4000000
TTL	11200

DNS Zones

- Each zone consists of Primary and Secondary Nameservers
- Each zone has its own zone file



Why do we need Zones

- DNS Zones are not necessarily physically separated from one another.
- All the information for a zone is stored in what’s called a DNS zone file
- Industry-specific TLD boost - It might be beneficial to your SEO to use top-level domains that are associated with your industry. For instance, those in the tech sector might get an SEO boost by using .tech or .io, as several startups already do. (SEO = Search Engine Optimization)
- Organizations can easily delegate control & access, so they don’t have to deal with Root providers like Internet Corporation for Assigned Names and Numbers (ICANN/IANA) etc.

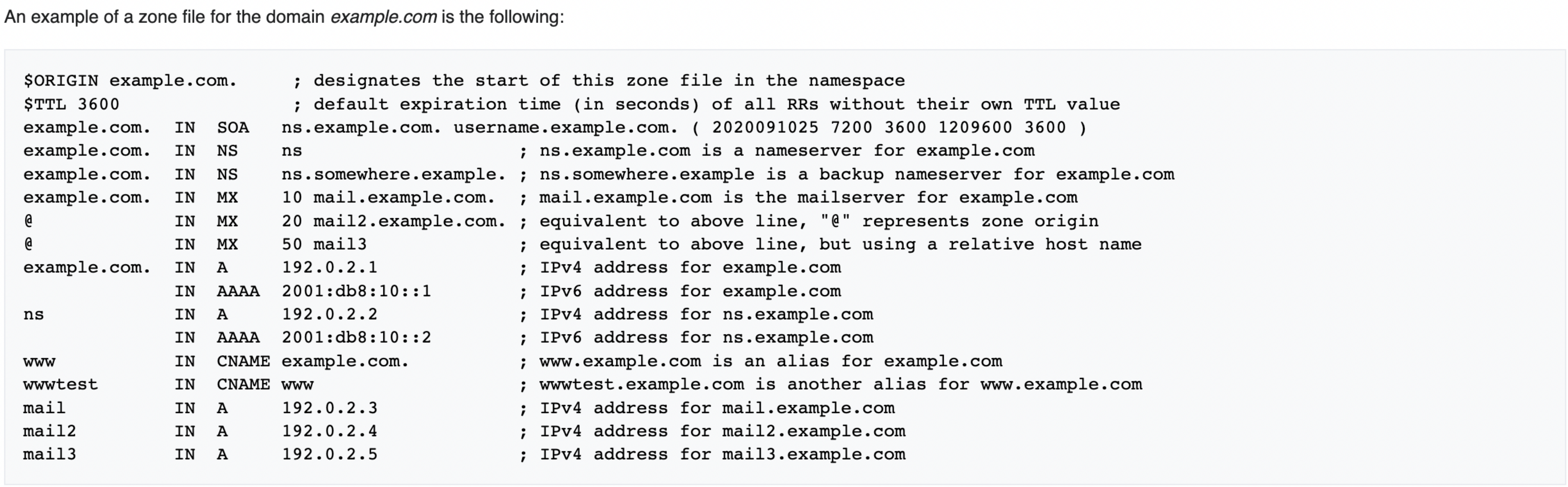
ICANN

DNS Zone Files

- DNS records (aka zone files) are instructions that live in authoritative DNS servers and provide information about a domain including what IP address is associated with that domain and how to handle requests for that domain.
- A zone file is a plain text file stored in a DNS server that contains an actual representation of the zone and contains all the records for every domain within the zone.
- Must start with SOA (START OF AUTHORITY) record.
- These records consist of a series of text files written in what is known as DNS syntax. DNS syntax is just a string of characters used as commands that tell the DNS server what to do.
- E.g: A records, AAAA (Quad A) records, MX Records, NS Records, PTR Records, CNAME, SRV, etc.

Example of Zone File

- This is how it looks like



An example of a zone file for the domain *example.com* is the following:

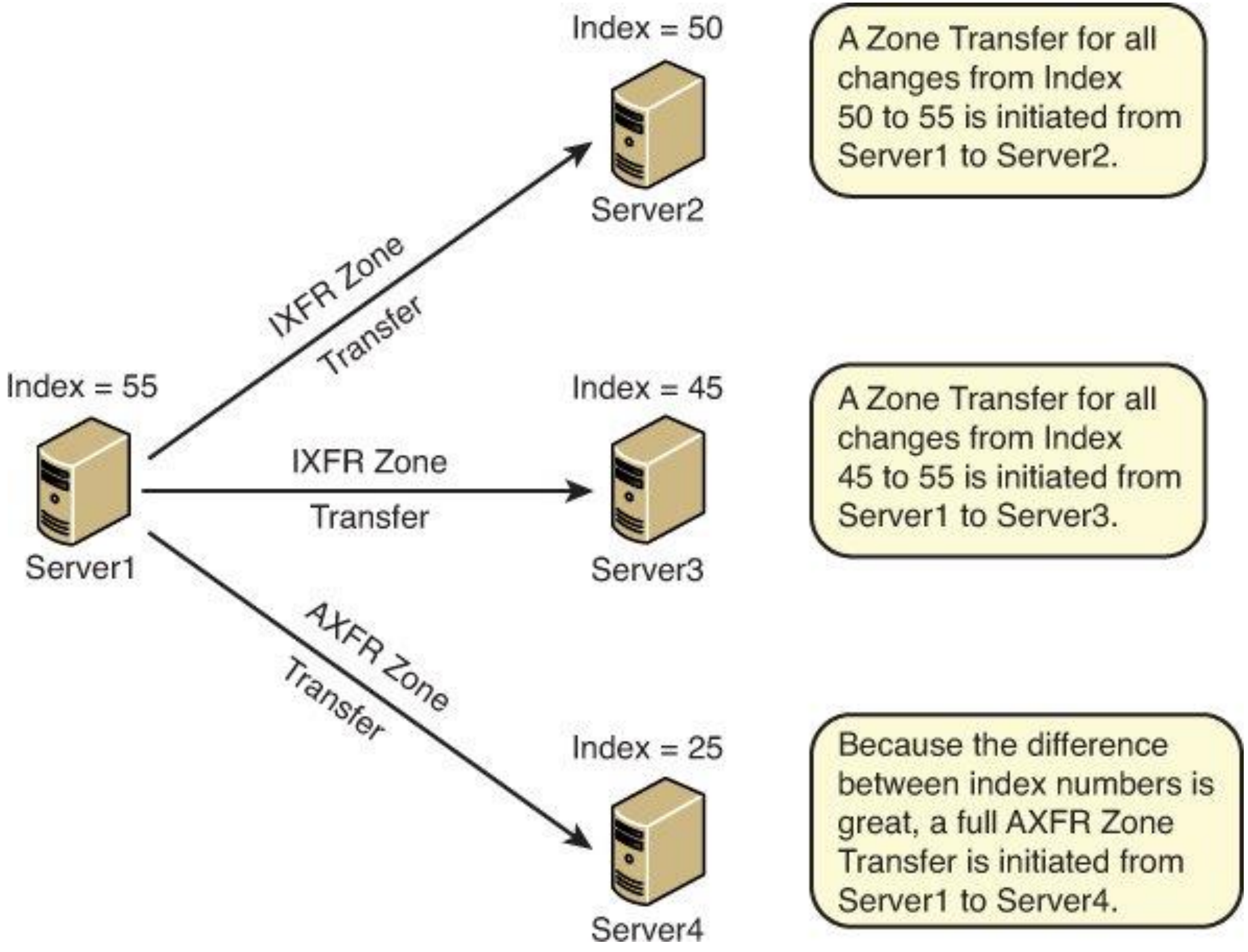
```
$ORIGIN example.com.     ; designates the start of this zone file in the namespace
$TTL 3600                ; default expiration time (in seconds) of all RRs without their own TTL value
example.com.  IN  SOA   ns.example.com. username.example.com. ( 2020091025 7200 3600 1209600 3600 )
example.com.  IN  NS    ns                    ; ns.example.com is a nameserver for example.com
example.com.  IN  NS    ns.somewhere.example. ; ns.somewhere.example is a backup nameserver for example.com
example.com.  IN  MX    10 mail.example.com.  ; mail.example.com is the mailserver for example.com
@             IN  MX    20 mail2.example.com. ; equivalent to above line, "@" represents zone origin
@             IN  MX    50 mail3              ; equivalent to above line, but using a relative host name
example.com.  IN  A     192.0.2.1             ; IPv4 address for example.com
              IN  AAAA  2001:db8:10::1        ; IPv6 address for example.com
ns            IN  A     192.0.2.2             ; IPv4 address for ns.example.com
              IN  AAAA  2001:db8:10::2        ; IPv6 address for ns.example.com
www           IN  CNAME example.com.          ; www.example.com is an alias for example.com
wwwtest       IN  CNAME www                   ; wwwtest.example.com is another alias for www.example.com
mail          IN  A     192.0.2.3             ; IPv4 address for mail.example.com
mail2         IN  A     192.0.2.4             ; IPv4 address for mail2.example.com
mail3         IN  A     192.0.2.5             ; IPv4 address for mail3.example.com
```

Structure of a Zone file

- DNS Zone files start with two mandatory records:
- Global Time to Live (TTL), which specifies for how records should be kept in local DNS cache.
- Start of Authority (SOA) record—specifies the primary authoritative name server for the DNS Zone.
- After these two records, the zone file can contain any number of resource records, which can include:
- Name Server records (NS)—specifies that a specific DNS Zone, such as “example.com” is delegated to a specific authoritative name server
- IPv4 (A)—a hostname and its IPv4 address.
- IPv6 (AAAA)—a hostname and its IPv6 address.
- (CNAME)—points a hostname to an alias. This is another hostname, which the DNS client is redirected to
- Mail exchanger record (MX)—specifies an SMTP email server for the domain.

Zone Transfer

- In DNS, you can copy data from the Master DNS zone to the Secondary DNS zones through a process called DNS Zone transfer. There are two types of zone transfer:
- **Full zone transfer (AXFR)** – a complete zone transfer, where the Secondary DNS servers copy the whole zone file.
- **Partial zone transfer (IXFR)** – In this case, the Secondary servers will check all the new changes that happened since their last update (deleted and added DNS records) and get only them.
- To function correctly, the system needs to keep been updated. That could happen in two ways:
- **Push** – The Master DNS server can propagate a zone transfer to the Secondary DNS servers.
- **Pull** – The Secondary DNS server can check for changes inside the Master zone, and if they find any differences by comparing the SOA records, they can start a zone transfer.



A Zone Transfer for all changes from Index 50 to 55 is initiated from Server1 to Server2.

A Zone Transfer for all changes from Index 45 to 55 is initiated from Server1 to Server3.

Because the difference between index numbers is great, a full AXFR Zone Transfer is initiated from Server1 to Server4.

Types of Authoritative Servers

- There are two types of authoritative servers: master (primary) and secondary. Each zone must have only one master name server, and it should have at least one secondary name server for backup purposes to minimize dependency on a particular node. Calling a particular name server a master or secondary server is misleading. Any given name server can take on either or both roles, as defined by the conf file.
- The zone data updates and maintenance are reflected in the master name server and the changes are then reflected in secondary name servers. Both master and secondary name servers are authoritative for a zone.
- E.g in your Router Config you usually set it to

  Primary: 8.8.8.8

  Secondary: 8.8.4.4

Use cases: Load Balancing and maximizing availability

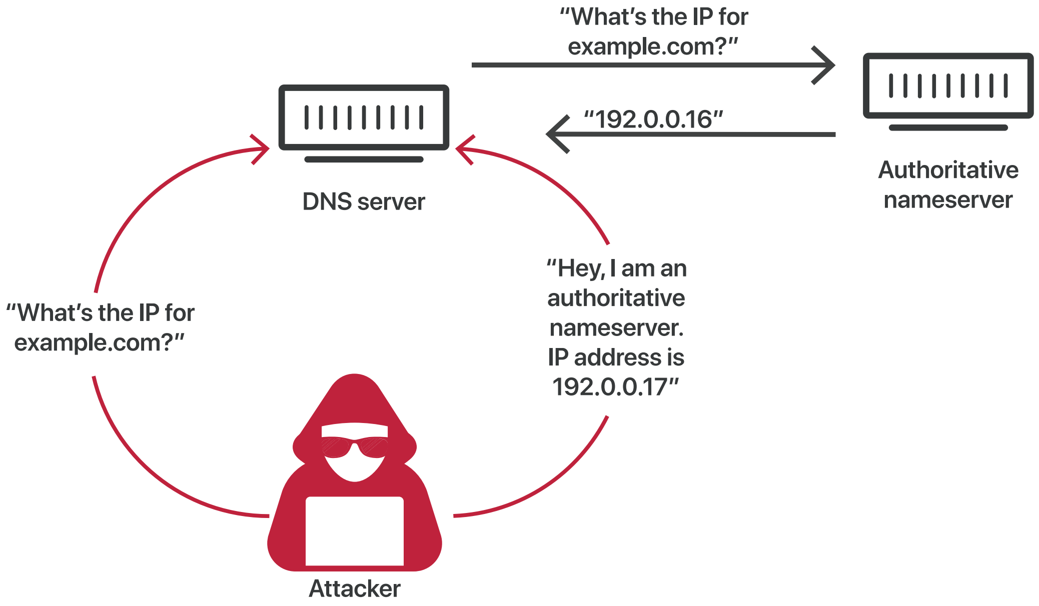
Dynamic DNS(DDNS)

- The rapid growth of the web and home computers with Internet access created a shortage of available IP addresses. This led to the Dynamic Host Configuration Protocol (DHCP), which lets ISPs assign IPs to their users dynamically.
- Dynamic DNS is a method of automatically updating a name server in the Domain Name System, often in real time, with the active DDNS configuration of its configured hostnames, addresses or other information.
- E.g: https://www.noip.com/

DNS Attacks

Some common DNS Attacks

- DNS spoofing/cache poisoning
- DNS Hijacking
- DNS Tunnels


"What's the IP for example.com?"
"192.0.0.16"
DNS server
Authoritative nameserver
"What's the IP for example.com?"
"Hey, I am an authoritative nameserver. IP address is 192.0.0.17"
Attacker

DNS Attacks..

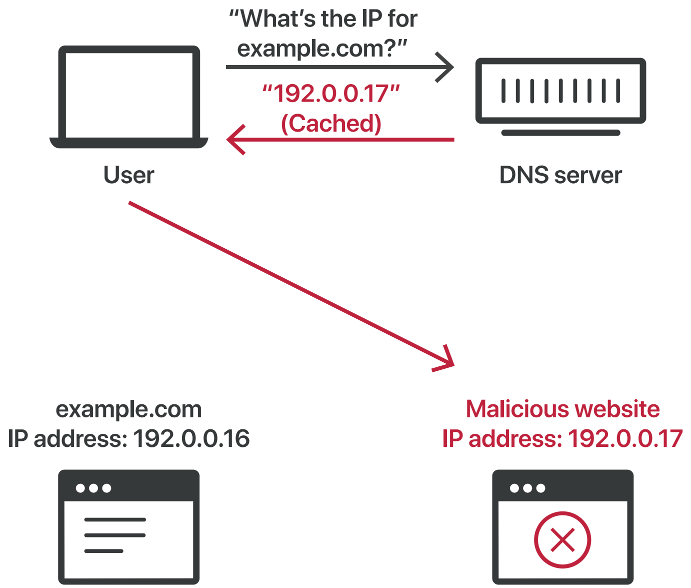
- DNS hijacking: In DNS hijacking the attacker redirects queries to a different domain name server. This can be done either with malware or with the unauthorized modification of a DNS server. Although the result is similar to that of DNS spoofing, this is a fundamentally different attack because it targets the DNS record of the website on the nameserver, rather than a resolver's cache.
- DNS tunneling: This attack uses other protocols to tunnel through DNS queries and responses. Attackers can use SSH, TCP, or HTTP to pass malware or stolen information into DNS queries, undetected by most firewalls.

DNS Cache Poisoning

- Attackers can poison DNS caches by impersonating DNS Nameservers, making a request to a DNS resolver, and then forging the reply when the DNS resolver queries a nameserver.

This is possible because DNS servers use UDP instead of TCP, and because currently there is no verification for DNS information.

Instead of using TCP, which requires both communicating parties to perform a 'handshake' to initiate communication and verify the identity of the devices, DNS requests and responses use UDP, or the User Datagram Protocol. With UDP, there is no guarantee that a connection is open, that the recipient is ready to receive, or that the sender is who they say they are. UDP is vulnerable to forging for this reason – an attacker can send a message via UDP and pretend it's a response from a legitimate server by forging the header data.



Enter DNSSEC

- **What is DNSSEC?**
- DNS Security Extensions (DNSSEC) is a security protocol created to mitigate this problem. DNSSEC protects against attacks by digitally signing data to help ensure its validity. In order to ensure a secure lookup, the signing must happen at every level in the DNS lookup process.

Dnssec continued….

- DNSSEC implements a hierarchical digital signing policy across all layers of DNS. For example, in the case of a ‘google.com’ lookup, a root DNS server would sign a key for the .COM nameserver, and the .COM nameserver would then sign a key for google.com’s authoritative nameserver.
- While improved security is always preferred, DNSSEC is designed to be backwards-compatible to ensure that traditional DNS lookups still resolve correctly, albeit without the added security. DNSSEC is meant to work with other security measures like SSL/TLS as part of a holistic Internet security strategy.
- DNSSEC creates a parent-child train of trust that travels all the way up to the root zone. This chain of trust cannot be compromised at any layer of DNS, or else the request will become open to an on-path attack.

- END OF PRESENTATION ANY QUESTIONS? :)