DNS

Before DNS: /etc/hosts distributed over FTP

Goals:

- Consistent namespace (Label → Data)
- Fast
- Reliable
- Decentralized

Fundamental Unit of Data: Resource Record

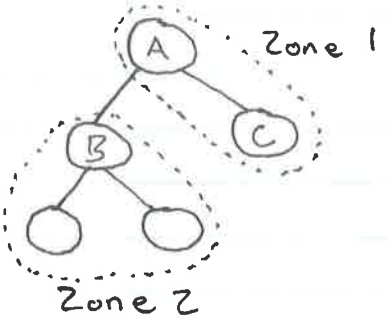
Consists of: Name, Type, Class, TTL, RDLength, RData
↑ IN (Class)   ↑ In seconds (TTL)

Example Types: A, AAAA
CNAME ("Canonical name" / alias)
MX
NS
PTR
TXT

Resource Records grouped into Zones

Each domain belongs to a zone. Subdomains are added to parent's zone or given own.

Authority over each zone granted to a single admin.

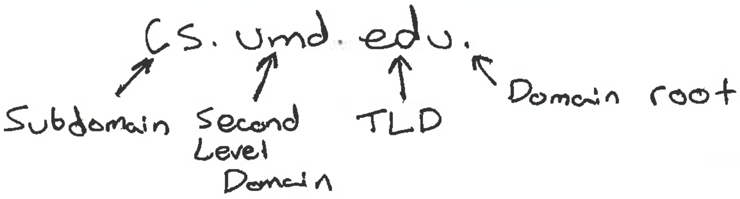


B is a subdomain of A, etc.

Zones are implemented by Name Servers

Multiple name servers per zone, and multiple zones per name server

Anatomy of a Fully Qualified Domain Name:



Each of these has a name server that knows about the name servers underneath it.

For example, the "edu" name servers know of the name servers for "umd.edu".

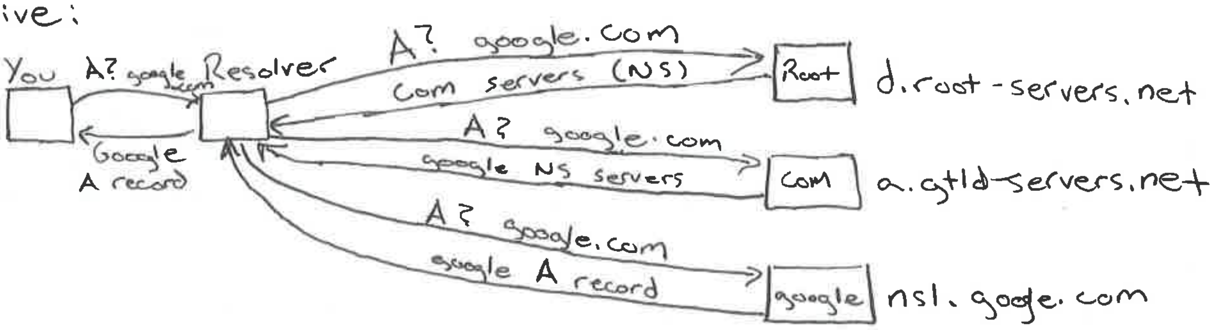
DNS Queries

0	15 16	31
QID		Flags
# Questions		# Answers
# Authority		# Additional
questions		
answers		
authority		
additional		

Recursion desired?
Recursion available?
Query / Response?

Suppose nothing is cached. How do we resolve the A record for google.com.?

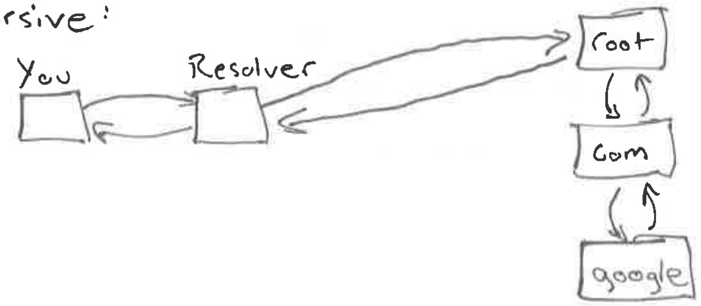
Iterative:



The root and TLD servers respond with Glue records. Why?

Both the NS record and the A record for the NS.

Recursive:

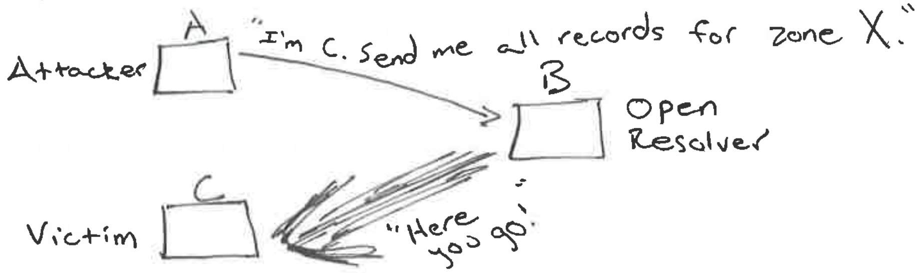


In reality, the resolvers caches aggressively.

Reverse lookup: PTR records under in-addr.arpa

Attacks:

Amplification



Cache Poisoning

Just one of many techniques.

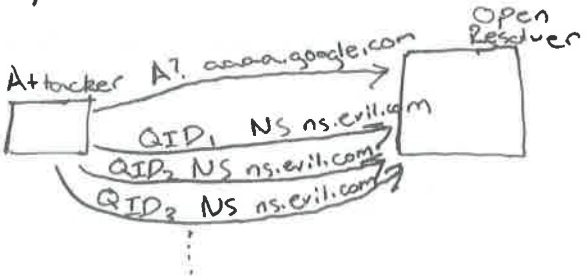
Attacker injects bad data into a resolver's cache.

Each query has:

QID : 2 bytes (assigned randomly)

Port : 2 bytes

Suppose server always listens on the same port:



Can poison the cache of Google's NS records!

Try to guess QID. If you fail, repeat with aaab.google.com, etc.