DNS

Domain Name System

www

.tech

IP Address

.com

.net

.gov

192.168.0.1

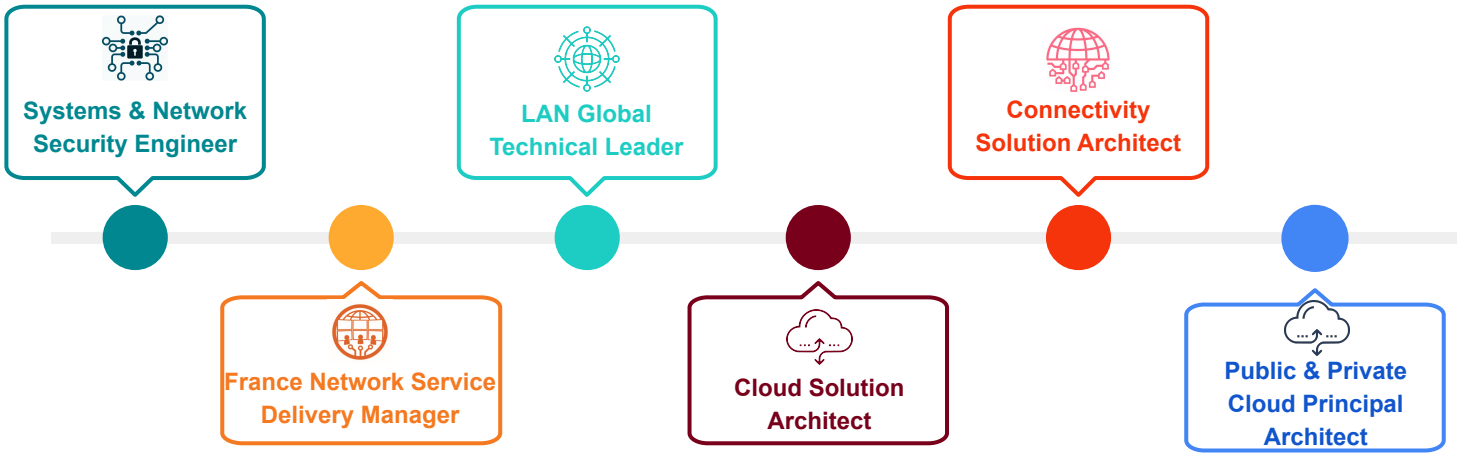
Julien MANTEAU - 2024

Objectives

- Describe the principles of the DNS
- Describe how the DNS data are handled and organized
- Identify the various types of name servers and how a DNS resolution is done
- Describe the key DNS operations
- Describe the role of DNS in killchain
- Identify the malicious usages of DNS

Teacher Presentation: Julien MANTEAU

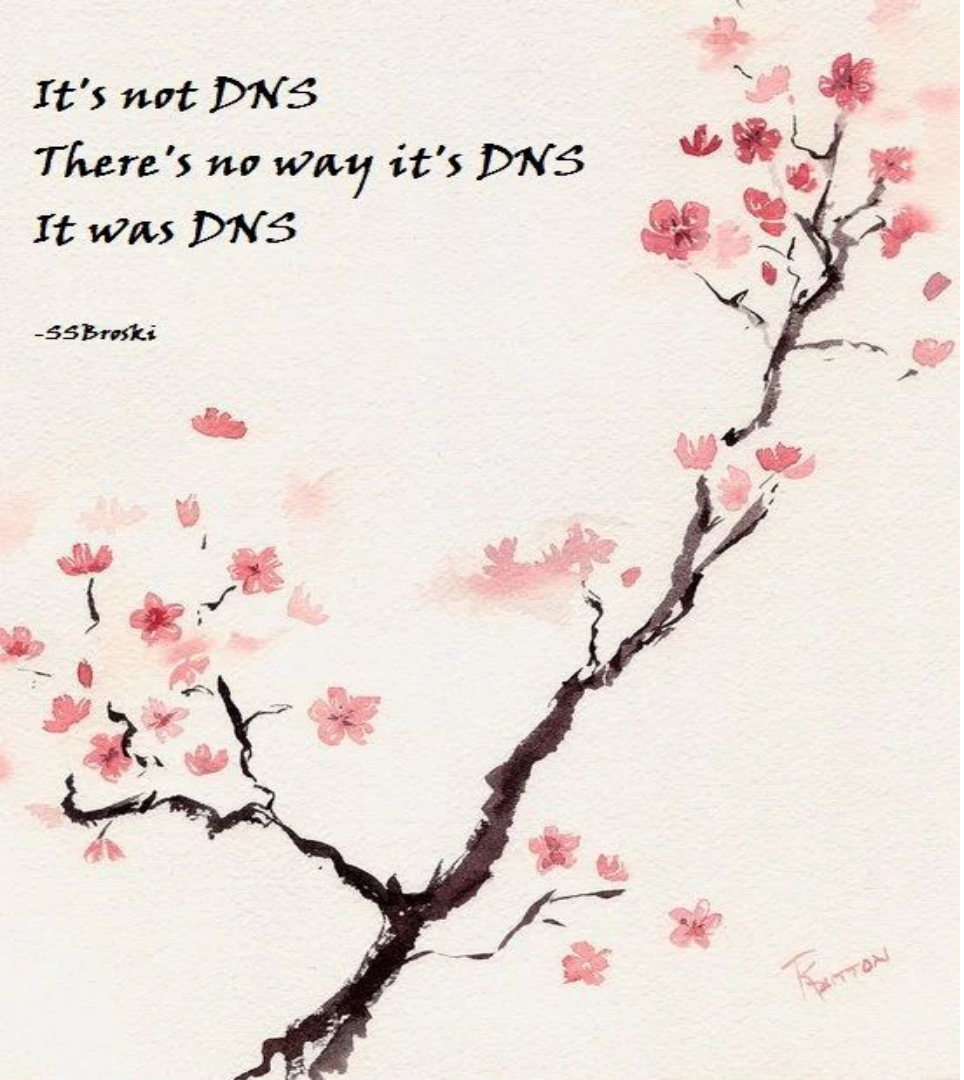
- Network & Telecom Engineer (ENSEEIHT)



- Lecturer at Bézier then Blagnac IUT in “Réseaux Télécoms” from 2010 to 2020
- Leader on Airbus CyberDiploma cursus setup and teacher/lecturer since 2021
- Hobbies:

AGENDA

1. **DNS Overview**
   What is the Domain Name System ?
2. **DNS & CyberSecurity**
   How DNS can be subverted ?
3. **Labs**



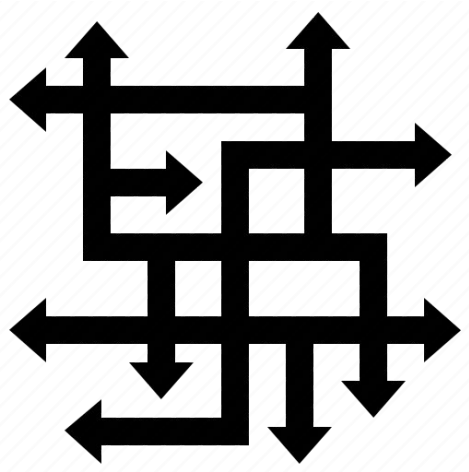
DNS Overview

DNS is hard ?

The only three hard problems in computer science are **cache invalidation** and **naming things.**

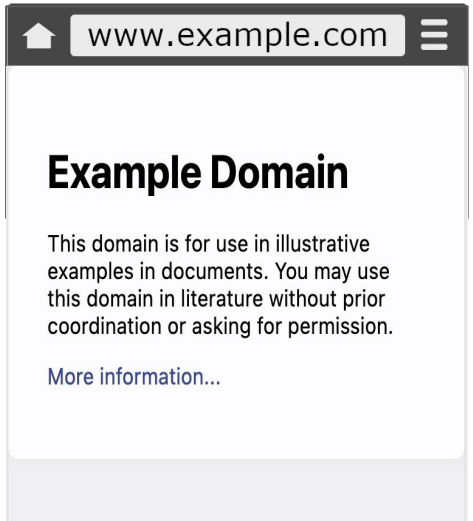
And **DNS is a caching system for names of things.**

(And off-by-one errors)



Question

What is happening behind the scenes ?



What do you know about the protocol allowing the name to be translated into technical information ?

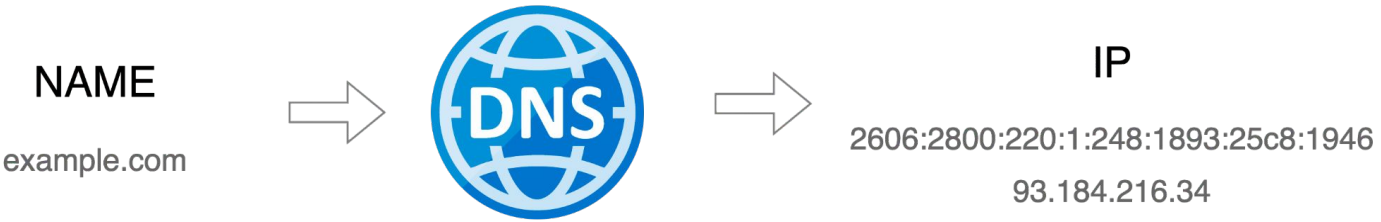
What is DNS ?

DNS is a distributed, hierarchical system for translating objects.

The main usage is Name to IP translation but it can also be IP to Name, Name to Ressources (Mail Servers, Domain Verification, etc)

DNS can mean the system (Domain Name System) or the infrastructure supporting it (Domain Name Servers)

DNS is a critical piece of the Internet infrastructure and any network infrastructure.

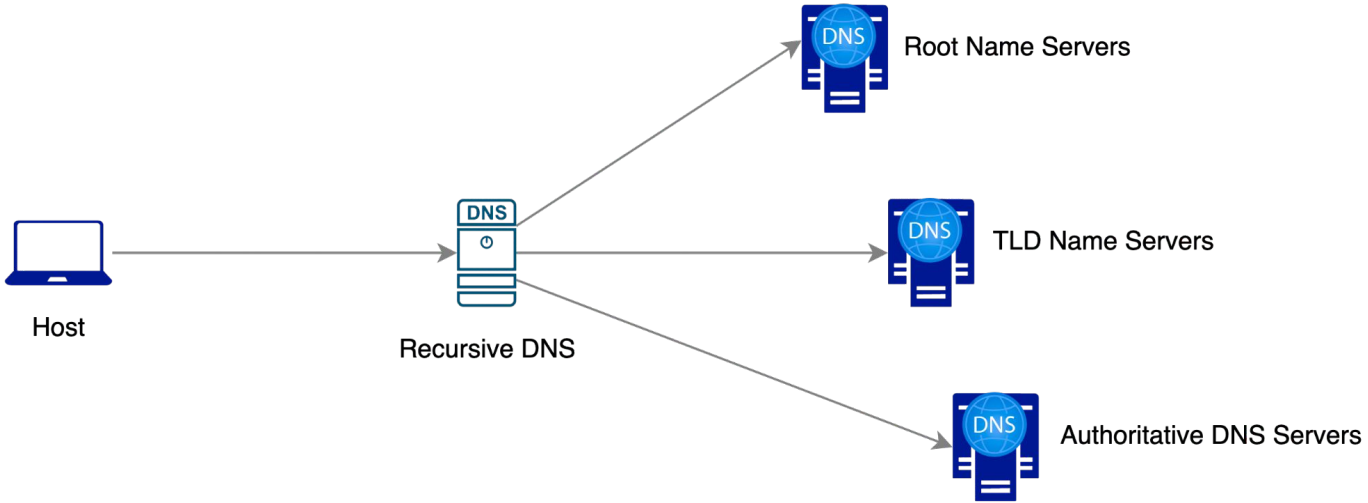


DNS Features

Globally Distributed	Multiple DNS servers managed by different operators	
Loosely coherent	Even if distributed, servers are still part of one global DNS system	
Scalable	The system can be scaled up and multiple servers can be added	
Reliable	Failure tolerance due to hierarchical distributed architecture and local caches	
Dynamic	Anyone can add domains and records without causing outage	

DNS Protocol

- DNS is a client-server protocol
  - Client (resolvers) requests resolution for a **function** (IP, mail server …) of a **name**
  - DNS server responds with information about the record
- Requests and responses are normally sent via **UDP/53**
- Also uses **TCP/53** for large requests
  - Ex: Zone transfers, large answers with DNSSEC, etc

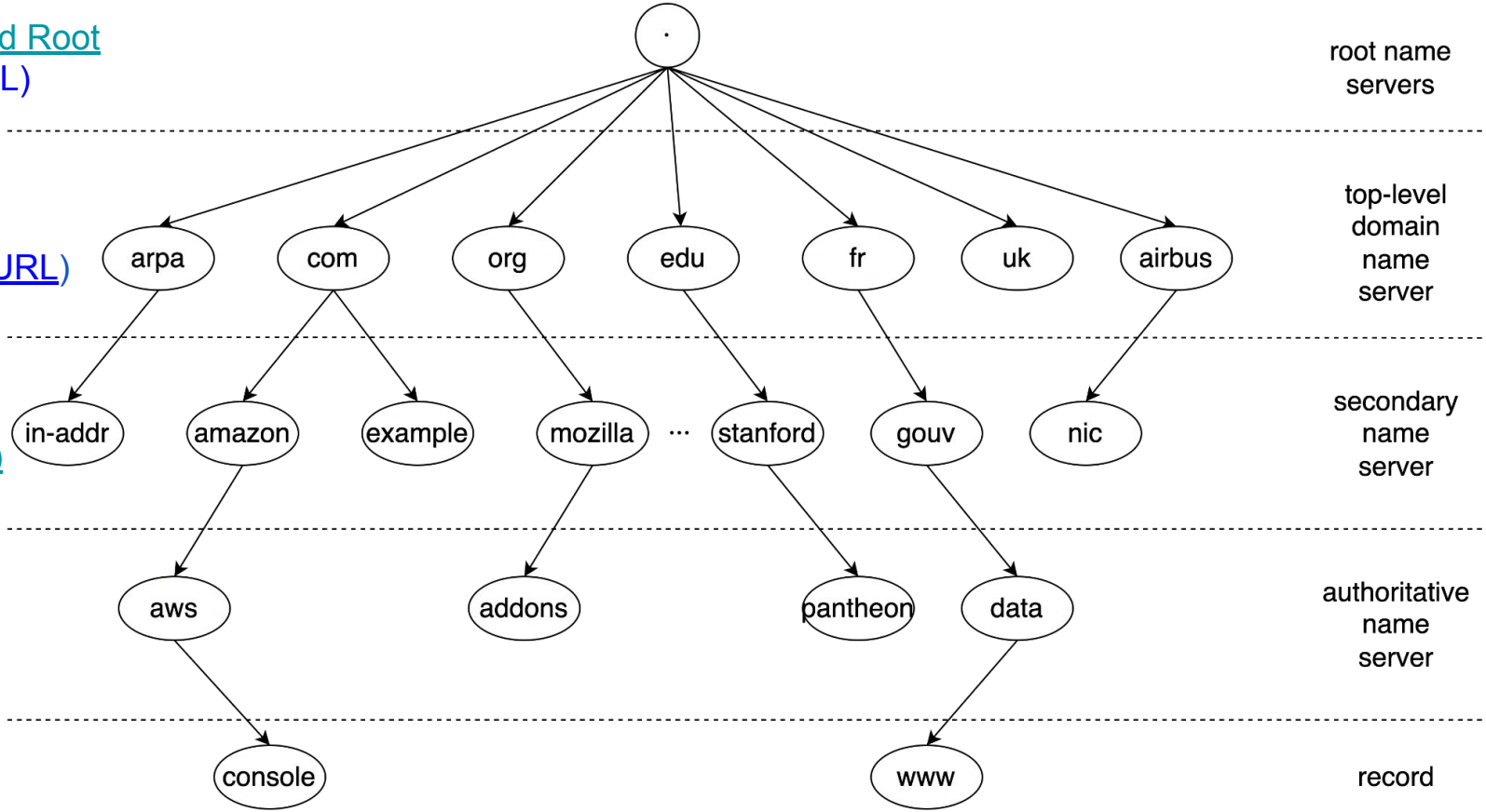


DNS Namespace Hierarchy

13 Anycasted Root Servers (URL)

1446 TLDs (Oct 2024) (URL)

~360 Millions domains (URL)



Who is responsible for DNS ?

- **ICANN (Internet Corporation for Assigned Names and Numbers)**: Oversees the global DNS structure, including the policies governing domain name registration.
- **IANA (Internet Assigned Numbers Authority)**: Part of ICANN, IANA coordinates key technical aspects of DNS, including the root zone file. It assigns IP address blocks, manages root servers, and designates top-level domains (TLDs).
- **Registry Operators:** Responsible for specific TLDs (like Verisign for .com), they maintain the records of domains under each TLD.
- **Registrars:** Accredited companies (e.g., GoDaddy) that allow individuals and organizations to register domain names.
- **ISPs and Network Admins**: Provide local DNS servers, translating domain names to IP addresses for end-users.

IANA type of TLDs

country-code top-level domains (ccTLD)

Two-character domains established for countries or territories. Example: .us for United States. .ca, .uk, .ae, and .fr are all ccTLDs.

internationalized country code top-level domains (IDN ccTLD)

ccTLDs in non-Latin character sets (e.g., Arabic or Chinese).

generic top-level domains (gTLD)

Top-level domains with three or more characters which include the generic .com, .net, .gov, .org, and more.

unsponsored top-level domains

Domains that operate directly under policies established by ICANN processes for the global Internet community, for example "edu".

sponsored top-level domains (sTLD)

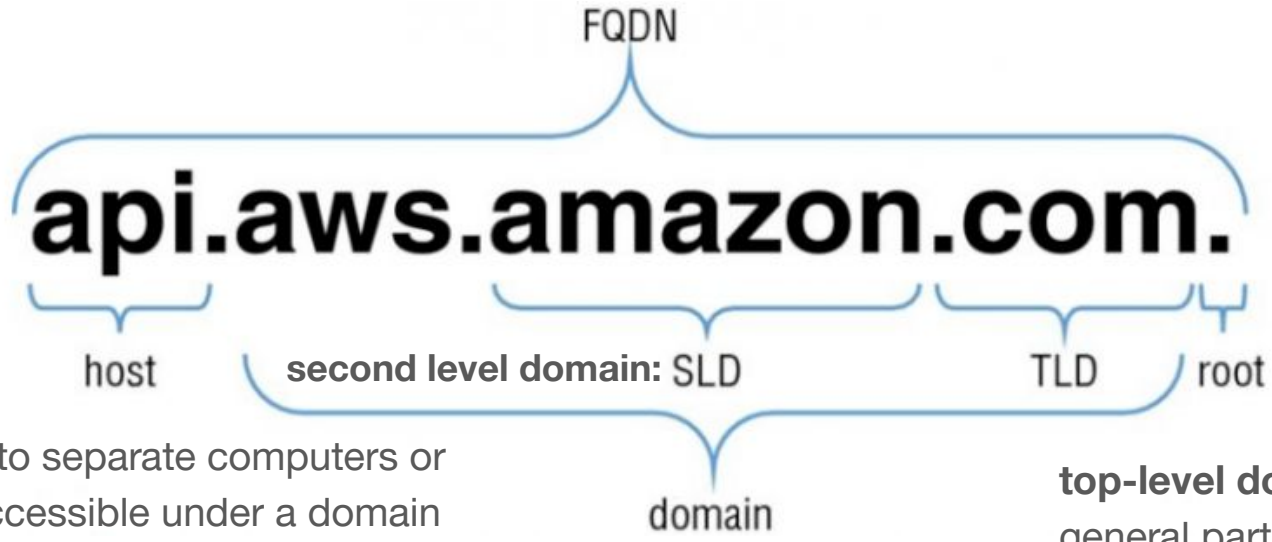
These domains are proposed and sponsored by private organizations that decide whether an applicant is eligible to use the TLD, based on community theme concepts. Include .airbus, .museum, .travel, etc

Special-Use top-level domain

This group consists of Address and Routing Parameter Area (ARPA), Private TLD (corp, local, etc) or specific TLD such as TOR .onion

DNS Terminology: “name” decomposition

A **Fully Qualified Domain Name** is an absolute name containing the host and the domain. **A proper FQDN ends with a dot**, indicating the root of the DNS hierarchy.



Root domain is the highest domain : “.”

Host refer to separate computers or services accessible under a domain

top-level domain is the most general part of the domain

The **domain** part of a FQDN. This is usually the part that needs to be loaned/registered to be used

Types of DNS Record

They are 47 active DNS types and 43 obsoleted ones. (List here). The most common are:

A **AAAA**	Links the hostname to the IP address. (A: IPv4, AAAA: IPv6)
CNAME	Point a Domain Name to another Domain Name
TXT	Text records are used to store descriptive text (site verification, SPF/DMARC mail record, etc)
MX	Mail Exchanger record point the **domain name** of the email server responsible for the domain name.
SOA	identify the Start of Authority with essential parameters of the zone (NS, contact info, default TTLs)
NS	NameServer records identify the responsible DNS Server for answering this zone.
PTR	Pointer record links an IP address to a domain name (opposite to A)
SRV	SeRVice record points to a host and its corresponding port for a specific service with a priority. It is used for discovering a services.

CAA, LOC, HINFO, RRSIG, NSEC, DNSKEY, GLUE are also seen but are more “advanced ones”

Quizz Time

www.pollev.com/jmanteau973

DNS Client: nslookup/dig

nslookup is officially deprecated in favour of **dig**.

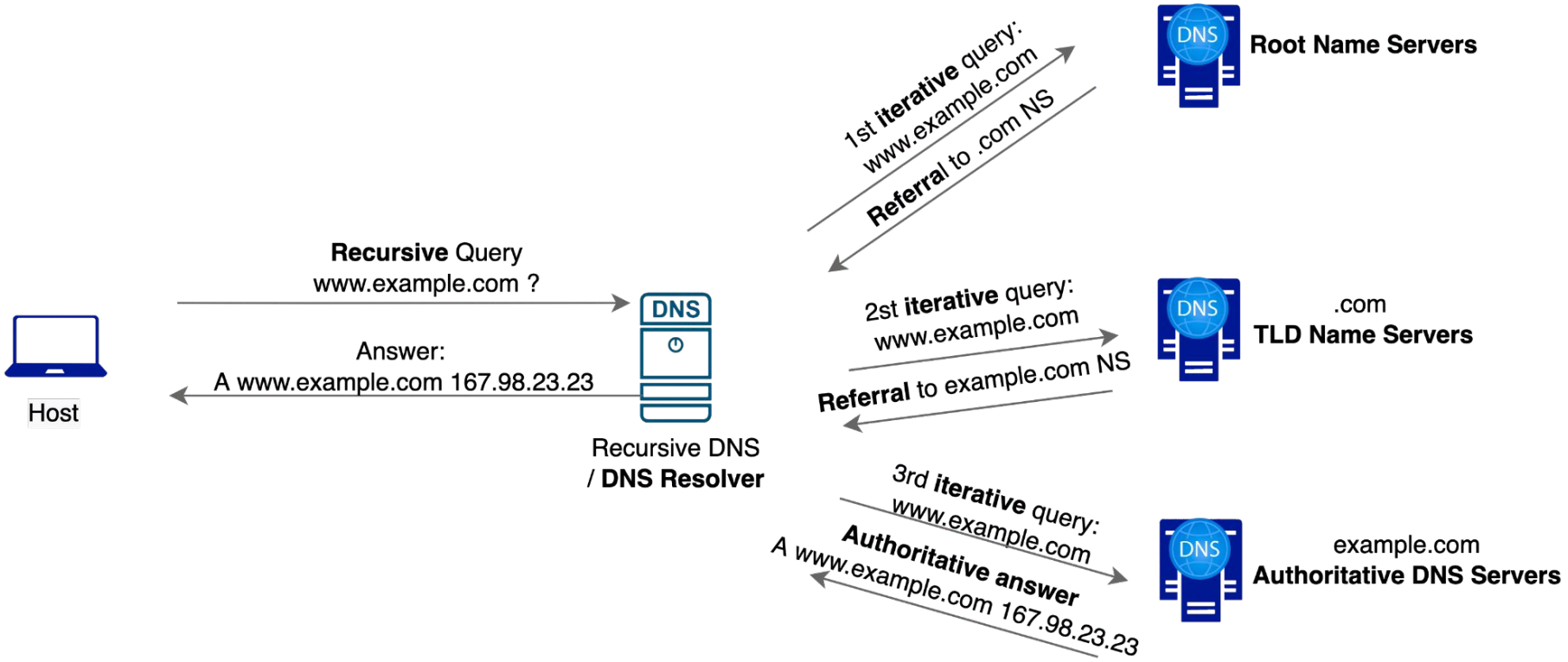
nslookup is however almost universally available (e.g. only resolver on Windows).

So my recommendation is:

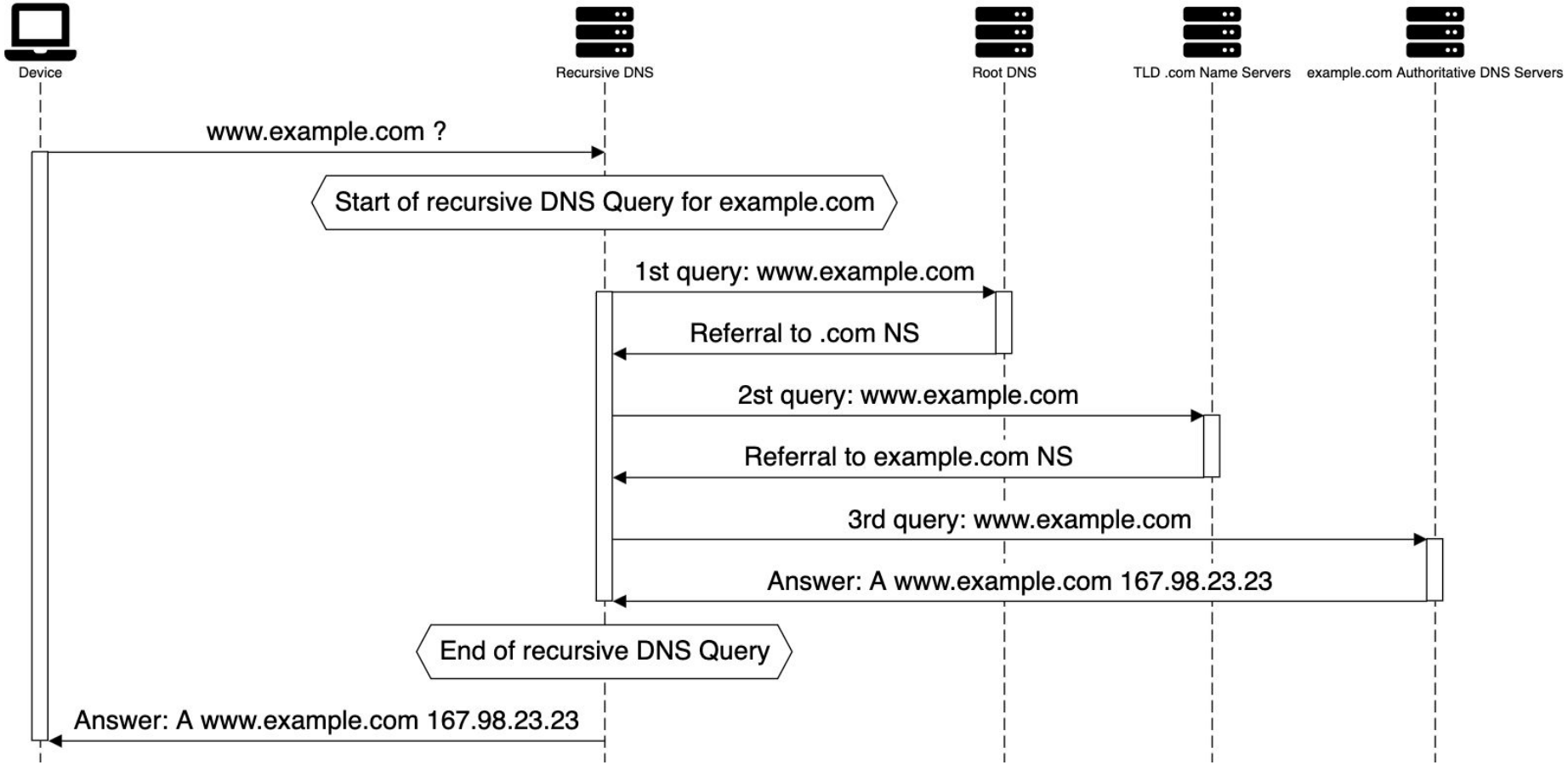
- Master and have dig on your system for dns investigation and testing.
- Know enough nslookup for basic troubleshooting on others devices

Practical Exercise: Request DNS with Dig

Full DNS Query and types of DNS Server



Full DNS Query



Lab: Full DNS Query sequence diagram for **www.impots.gouv.fr**

Practical Exercise: Full DNS Query

Based on the previous sequence diagram (click on link), please recreate the full chain of the resolution of www.impots.gouv.fr

```
❯ dig www.impots.gouv.fr +noall +answer
www.impots.gouv.fr.     129  IN   CNAME    www.impots.gouv.fr.edgesuite.net.
www.impots.gouv.fr.edgesuite.net. 21429   IN CNAME a1352.dscb.akamai.net.
a1352.dscb.akamai.net. 20   IN   A    95.100.133.81
a1352.dscb.akamai.net. 20   IN   A    95.100.133.74
```

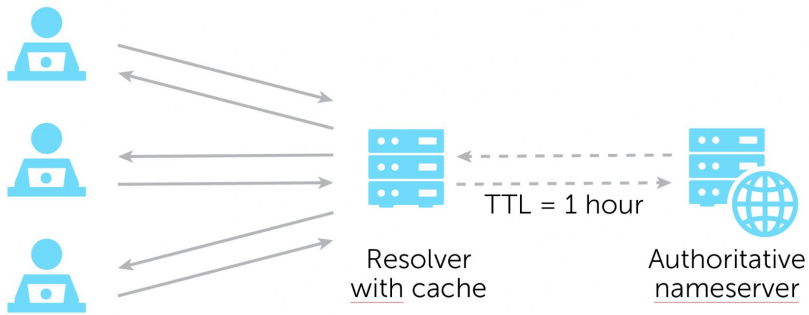
TTL: Definition

TTL (Time To Live) is a value configured on the DNS Server which informs the client / resolver **how long it should store the DNS record details before sending the query to get the updated information**. The DNS Server returns the TTL Value along the record.

Per the DNS RFC, DNS TTL values can be configured from 0 seconds to 248555 days.

However DNS client implementations can have different values.

This is why I perso



DNS Cache

DNS Cache (sometimes called a DNS resolver cache) is a temporary database, maintained by an OS, that contains records of all recent DNS lookups.

Before attempting to resolve a name, the OS will check if the entry is present in the cache. The entries are cleared according to their TTL. Before requesting he will also check the “static cache”: the hosts file.

A browser will also have a DNS cache that they will use before using the OS cache (and the default resolver if not in OS cache OR their own resolver)

	Windows	Linux (if present)	Mac (12.x)	Chrome	Firefox
Check Cache	ipconfig /displaydns	resolvectl query $HOSTNAME	Hashed as considered as private data	chrome://net-internals/#dns	about:networking#dns
Clear Cache	ipconfig /flushdns	resolvectl flush-caches	sudo dscacheutil -flushcache; sudo killall -HUP mDNSResponder		

TTL: Lookup

```
❯ dig wikipedia.org +noall +answer
wikipedia.org.          403     IN      A       185.15.58.224
```

ISP DNS Resolver Local TTL

```
❯ dig ns wikipedia.org +noall +answer
wikipedia.org.          2907    IN      NS      ns0.wikimedia.org.
wikipedia.org.          2907    IN      NS      ns2.wikimedia.org.
wikipedia.org.          2907    IN      NS      ns1.wikimedia.org.
```

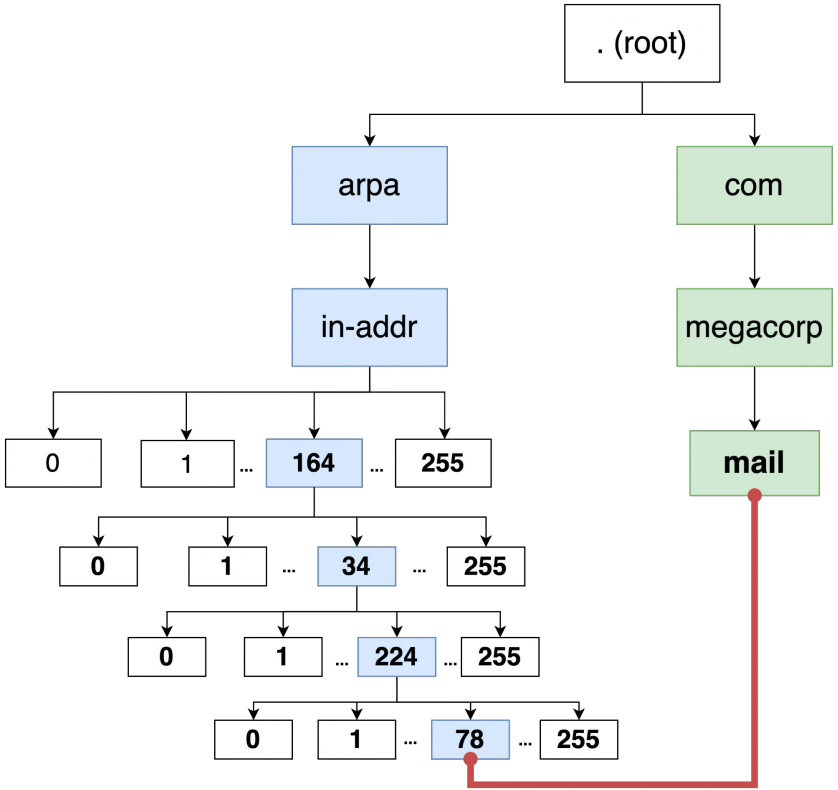
Locate and direct Query to authoritative NS

```
❯ dig wikipedia.org +norecurse @ns0.wikimedia.org +noall +answer
wikipedia.org.          600     IN      A       185.15.58.224
```

Authoritative TTL answer

If your default DNS server is not the authoritative server for the zone you are digging (so 99,99% of the time) dig will show the time remaining (until the next refresh) instead of the raw TTL value in this position.

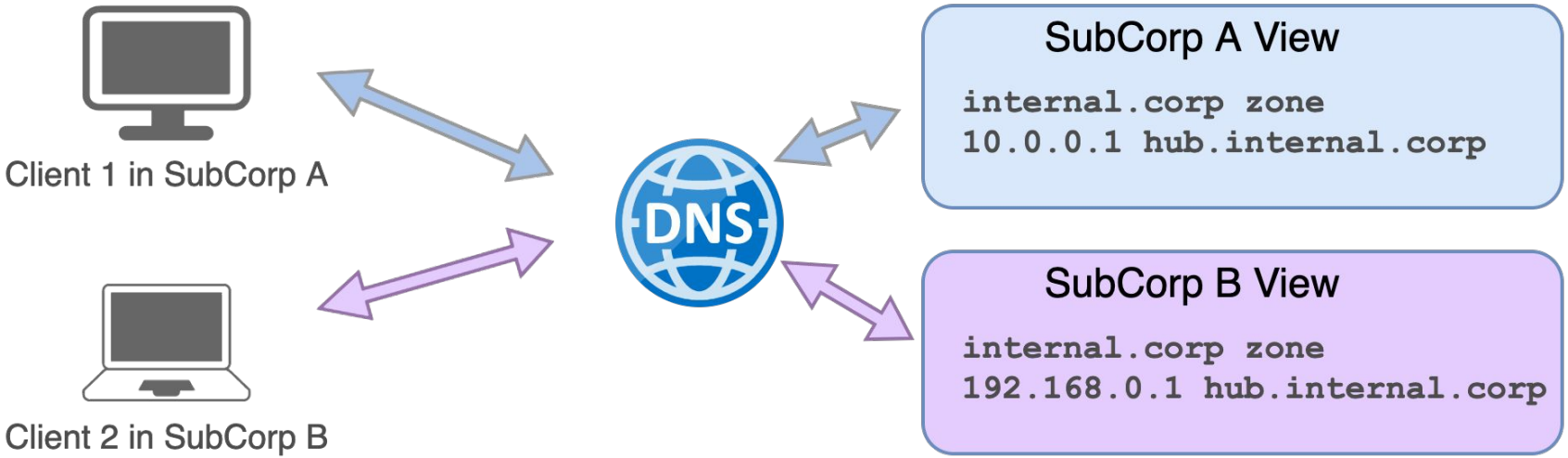
DNS Concepts: Reverse Mapping / DNS Lookup, PTR Record



`78.224.34.164.in-addr.arpa PTR mail.megacorp.com`

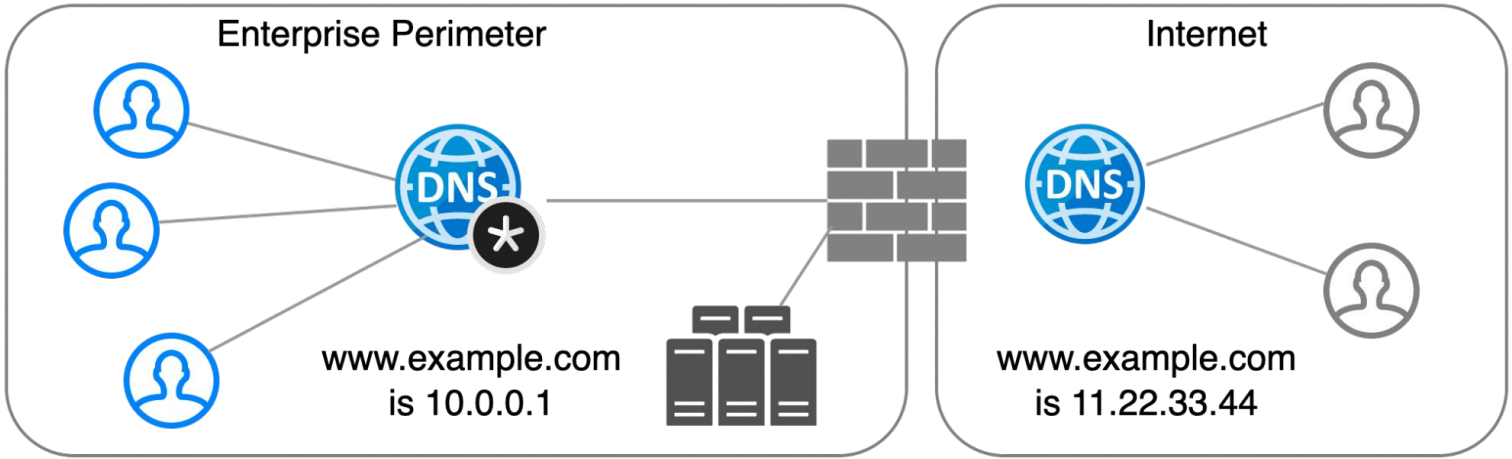
This is equal to :
"Pointer from IP 164.34.224.78 to mail.megacorp.com"

DNS Concepts: Views



The DNS Server matches the client subnet and choose the right view and answer from it. Here this is simple as the DNS Server as local zone present

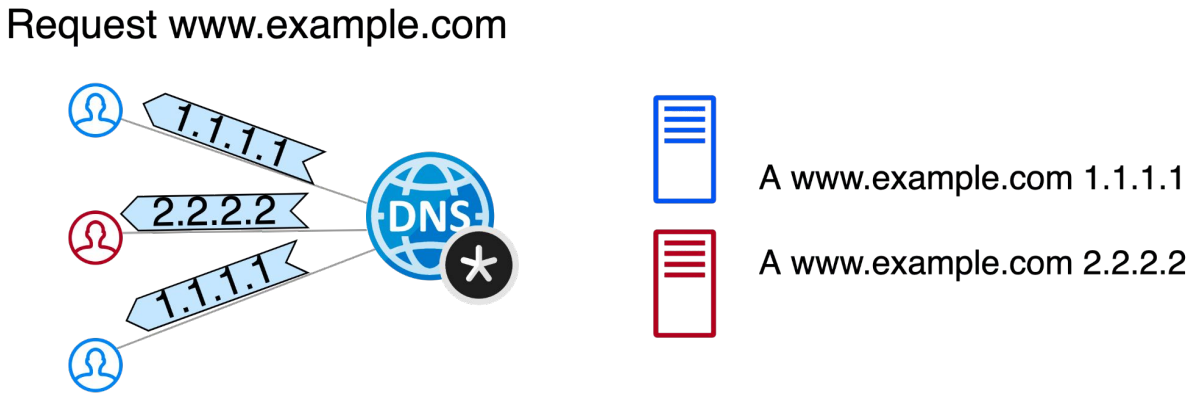
DNS Concepts: Split Horizon



Split Horizon is the DNS concept that the same resource is having different DNS entry depending from where the resolution is done. It can be implemented in different ways:

- DNS Views
- Different DNS servers (hence different zones)
- DNS Global Load Balancing
- RPZ

DNS Concepts: Round Robin

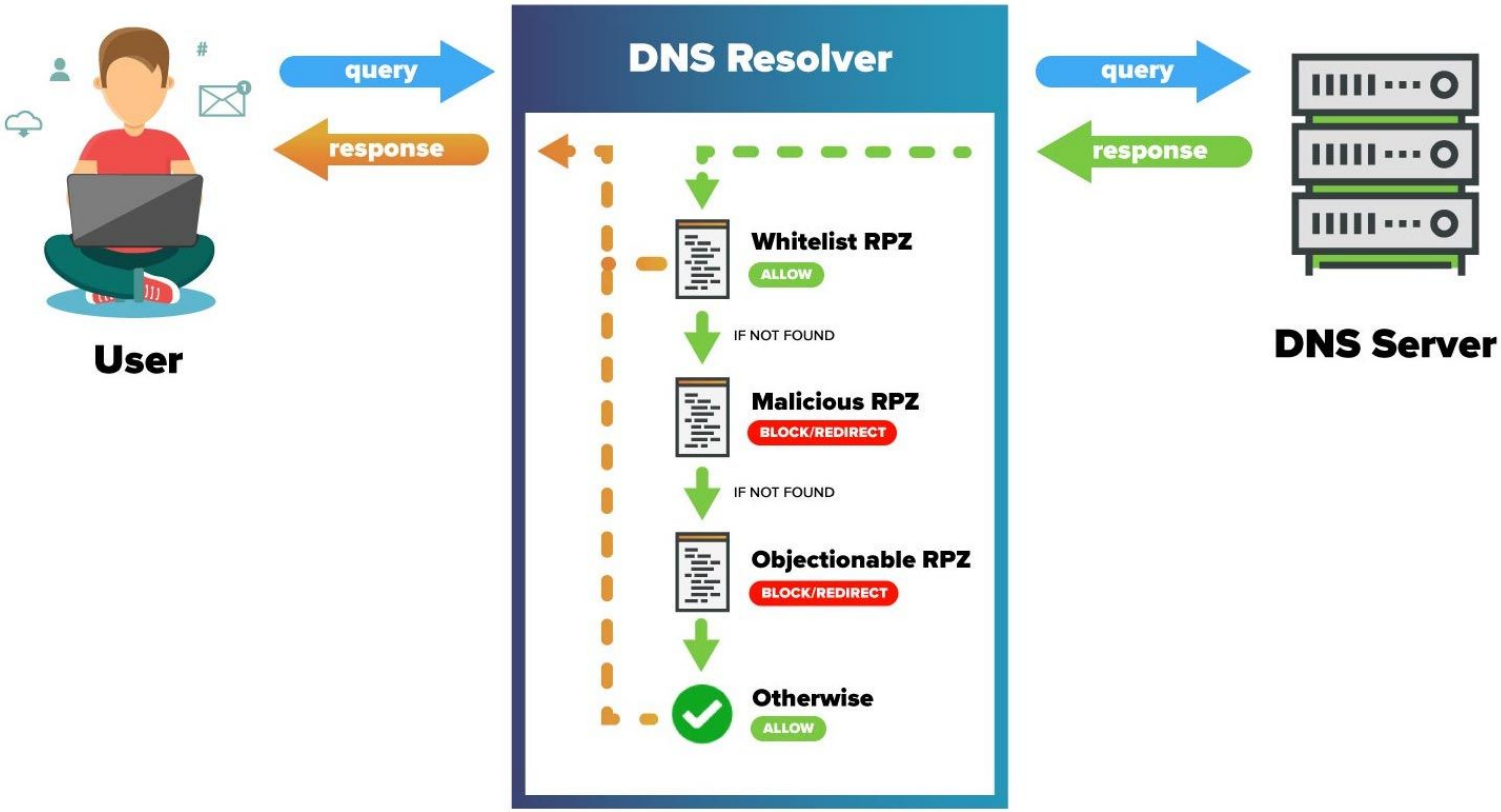


- **What it is**: A DNS load balancing method managed by the authoritative nameserver, without needing dedicated load-balancing hardware.
- **Use Cases**: Ideal for websites or services hosted on multiple redundant servers or when a single DNS name directs to multiple proxy servers.
- **How it works**: The nameserver provides a different IP address in each DNS response, rotating through available server IPs for balanced traffic distribution.

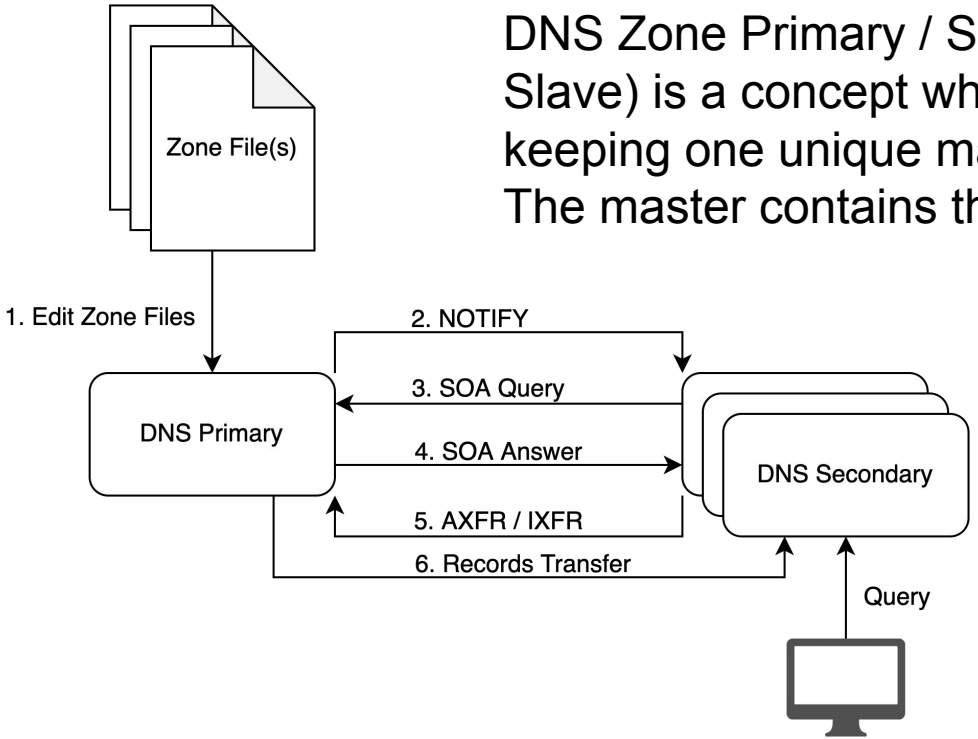
DNS Concepts: RPZ definition

- DNS Response Policy Zone (RPZ) is a DNS zone which allow to customize policy in DNS servers, so that the server returns modified answers to clients' queries.
- **RPZ** provides a way to **alter a DNS response** in real time. RPZ is a "lying" mechanism.
- It can be used to modify potentially unsafe DNS Data to block communication or provide local data to redirect to a "walled garden".
- As we can alter query response or block any domain using RPZ it is also known as **DNS firewall (or DNS Sinkhole)**.

DNS Concepts: RPZ schema



DNS Concepts: Primary, Secondary & Zone Update



DNS Zone Primary / Secondary (used to be called Master / Slave) is a concept which allow DNS servers to scale while keeping one unique management point for zone edition. The master contains the DNS data source of truth.

Secondary reads Master SOA Resource Record, check the serial number and request zone transfer if higher

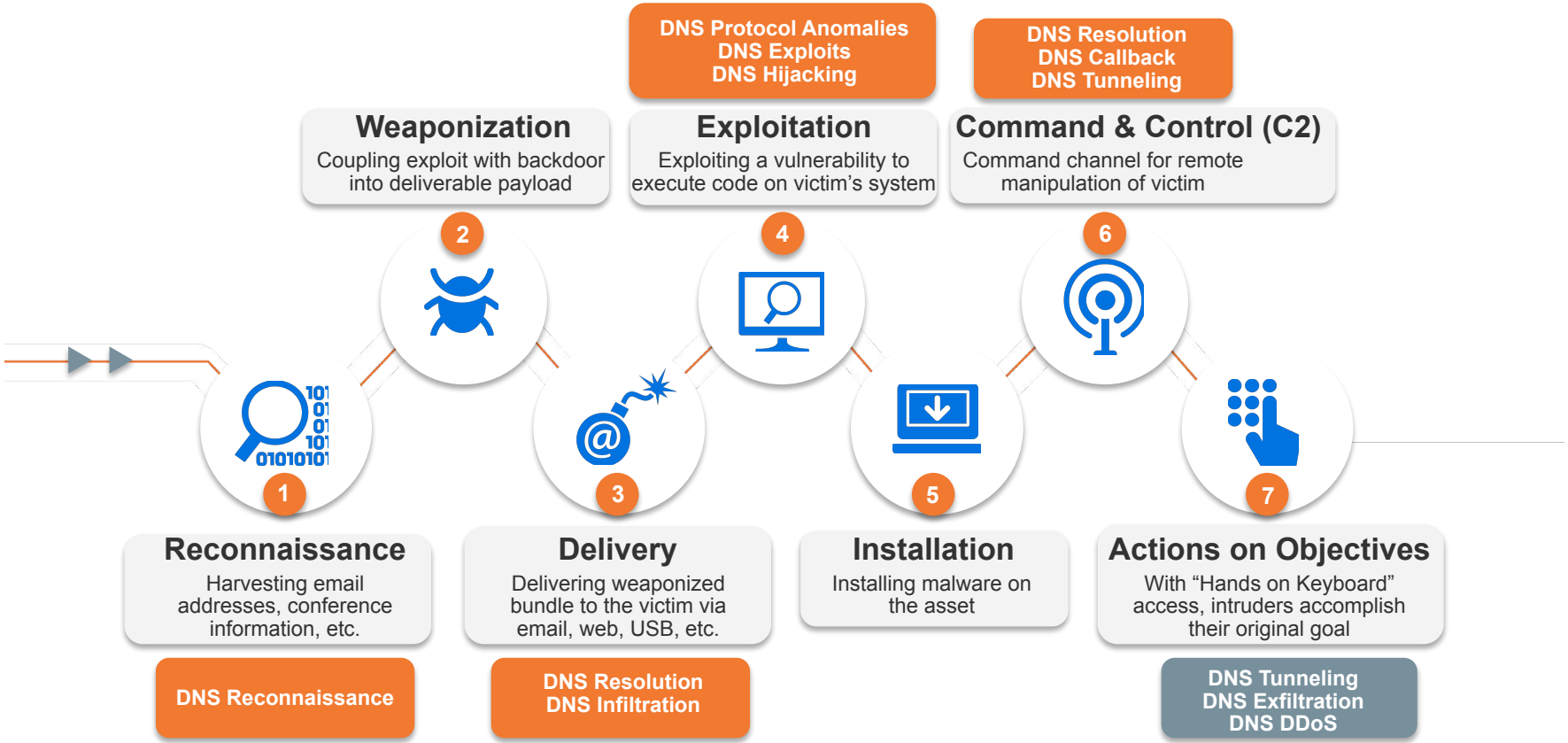
It is impossible to determine from a query result that it came from a zone primary or secondary.

Quizz Time

www.pollev.com/jmanteau973

DNS & CyberSecurity

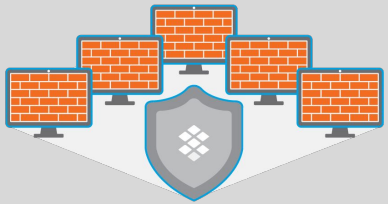
How DNS is used by malware (DNS usage in the KillChain) ?



1 Preventing communication over DNS

2 Stopping APTs/malware from using DNS

3 Defending DNS Infrastructure



1 Preventing communication over DNS

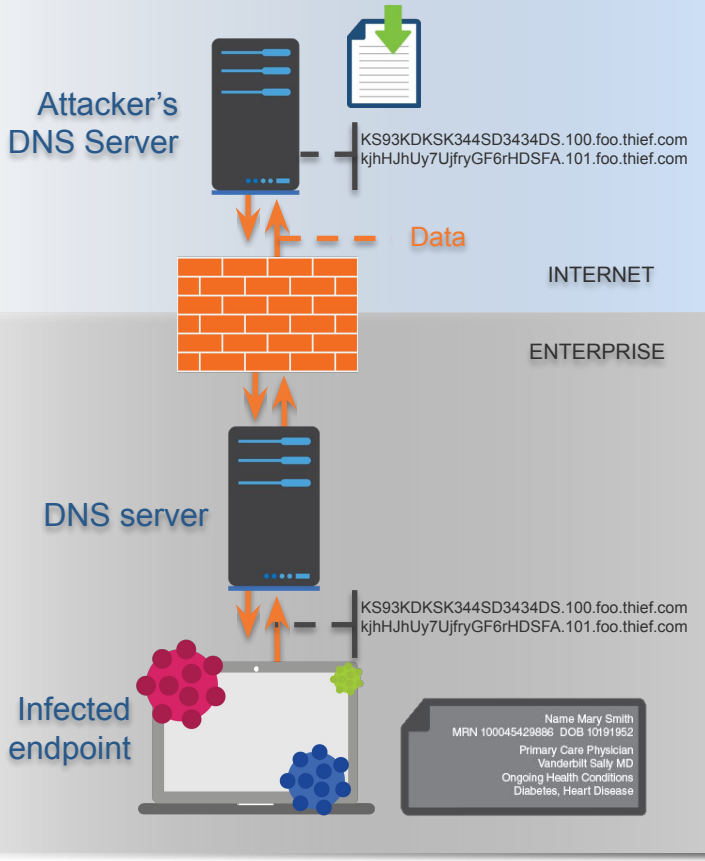
2 Stopping APTs/malware from using DNS

3 Defending DNS Infrastructure

DNS as a Transport Mechanism

Exfiltration

- Attacker registers a domain & sets up an authoritative DNS server on the Internet to act as the tunnel endpoint
- Data to be sent from inside network is:
  - Encrypted using public key
  - Encoded into a-z, 0-9 and – using algorithm such as Base32, which allows up to 110 bytes to be encoded into an FQDN
  - Divided into chunks of up to 63 characters (label limit)
  - Sent as individual queries in format of <chunk>.domain
- Queries can be sent direct (if allowed by firewall), via a network's existing DNS servers or via proxies (e.g. request http://<chunk>.domain)
- If sent via DNS the source address can be spoofed to make tracing harder
- Attacker's authoritative server receives encoded chunks, reassembles data, decodes & decrypts using private key



DNS as a Transport Mechanism

Infiltration

- Can be used to download a payload in a covert way for example.
- TCP features can be imitated by encoding the chunks with additional data, such as Checksum & Packet Number
- Data can be sent back in a variety of records, e.g.
  - A - allowing 4 bytes (enough for codes, e.g. 1.1.1.200 = resend packet 200)
  - AAAA - allowing 16 bytes
  - MX record : 2 bytes + domain name (255 bytes)
  - CNAME - allowing up to 110 bytes in Base32
  - TXT - allowing N x 220 bytes in Base64 (up to RDATA limit of 64kB)
  - NULL - allowing up to 256 bytes
- Using TXT and NULL make transmission faster, at expense of easier detection

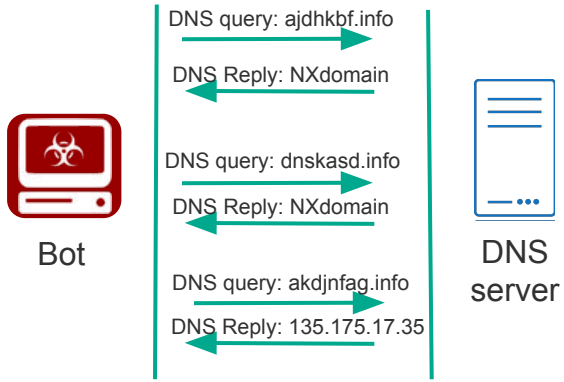
DGA – Domain Generation Algorithm

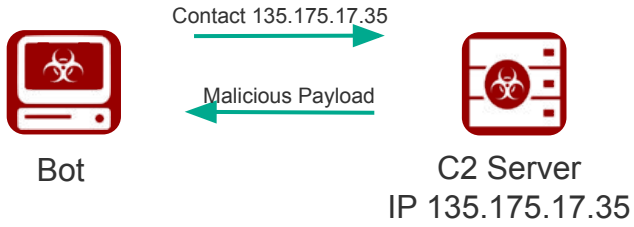
An algorithm producing Command & Control (C2) rendezvous points dynamically

For example: every day malware connects to time-based server FQDN: *<month>-<day>-<year>.com*

ie. on December 24, 2017 malware connects to 2017-12-24.com

Example Family	Example Domain
DirCrypt	vlbqryjd.com
Bamital	b83ed4877eec1997fcc39b7ae590007a.info
CCleaner	ab6d54340c1a.com





Lookalike Domain Detection

- Detects from Newly Observed Domains (NOD), domain names that try to mimic popular domains (Alexa top-100) or strategic domains for phishing (e.g. banks) and spamming.
- Detects domains that overlaps and/or have homograph n-grams, e.g. letter ‘O’ and number ‘0’ are considered homographs.
  Use distance analysis to detect the likelihood of lookalike attacks
- Detects
  - Letter replacement, example g00gle.com, goooogle.com, bankofthevvest.com, rn1cr0soft.com
    appIe.com >>> appIe.com (Arial) app1e.com >>> appIe.com (Cutive)
  - TLD change, example, walmart.cc
  - other generic highly resembled domains, example: bankofAmericas.com
- SOA record is analyzed and used for catching false positives.

Legitimate DNS Tunneling

- g63uar2ejiq5tlrkg3zezf2fkemc6pi88tz.er.spotify.com
- *a-0.19-b3000081.a010083.15e0.1d99.36d4.210.0.ic7arfsqqzf694fs8zf8nz2t9b.avts.mcafee.com*
- 4rzjp8zy7i7vawluximoxrko1p2tn58gj0fjjj2g.p.03.s.sophosxl.net
- p2.a22a43lt5rwfg.ihg5ki5i6q3cfn3n.191742.i1.ds.ipv6-exp.l.google.com
- a1294.w20.akamai.net
- dnn506yrbagrg.cloudfront.net
- A98dc034c7781a941ebabac02262202668bbe918ea9fb5289cd2.r58.cf2.rackcdn.com

C2 over DNS over HTTPS (DoH)

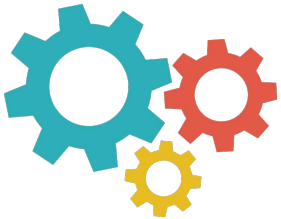
- Domain fronting

Domain fronting is a technique that masks the destination of a request by using different domains at various stages of the connection, allowing traffic to bypass certain filtering or detection systems.

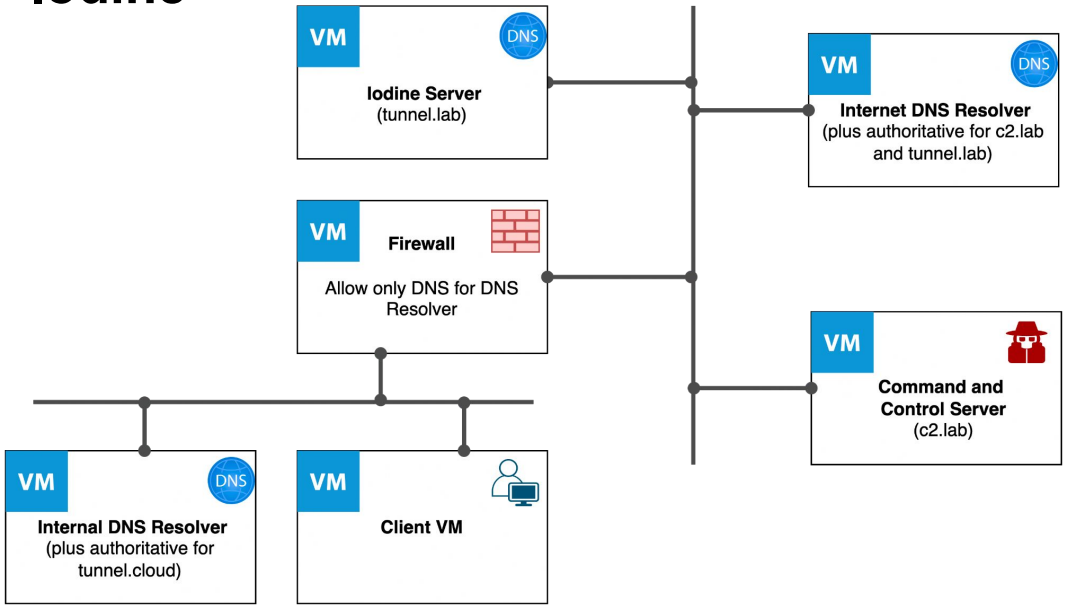
curl 'https://**google.com**/resolve?name=**malwareC2.com**'
-H "Host: **dns.google.com**"

Unencrypted DNS lookup & TLS SNI: **google.com**
Encrypted HTTP host: **dns.google.com**
Requested Domain: **malwareC2.com**

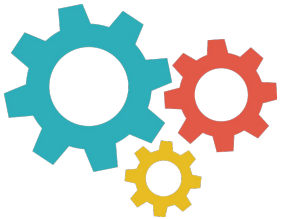
Lab: DNS Tunneling for bypassing filtering



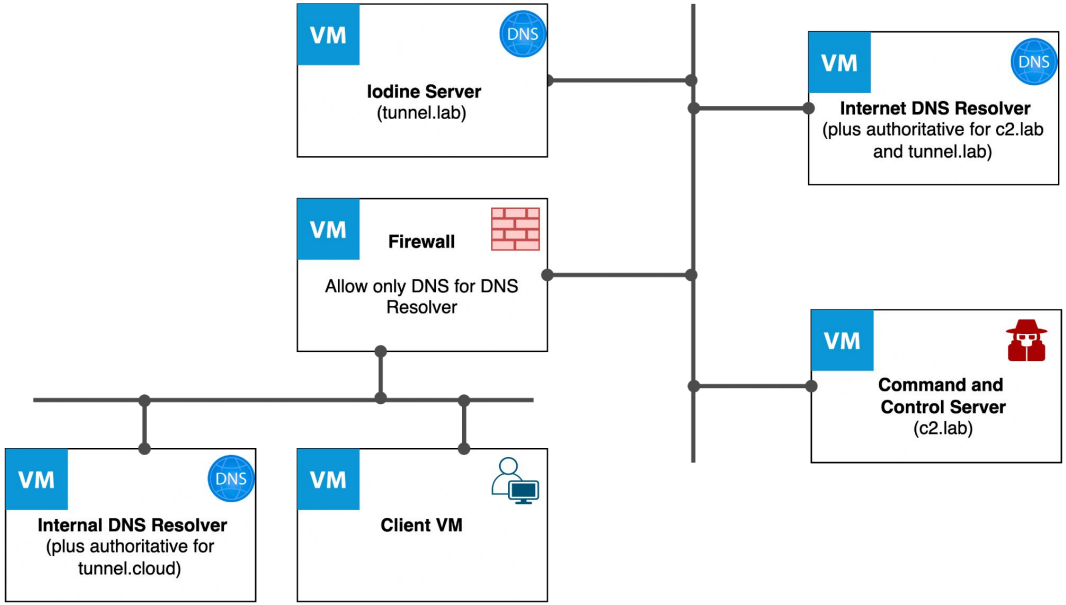
Practical Exercise: DNS Tunneling with Iodine



Lab: Command and Control over DNS



Practical Exercise: DNS C2 with Sliver



1 Preventing communication over DNS

2 Stopping APTs/malware from using DNS

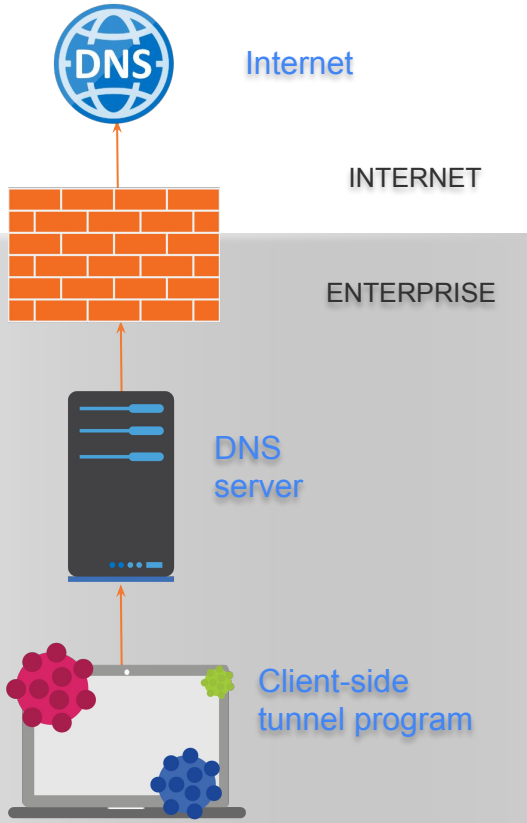
3 Defending DNS Infrastructure

Detecting DNS Tunnels & Data Exfiltration

Traditional approach

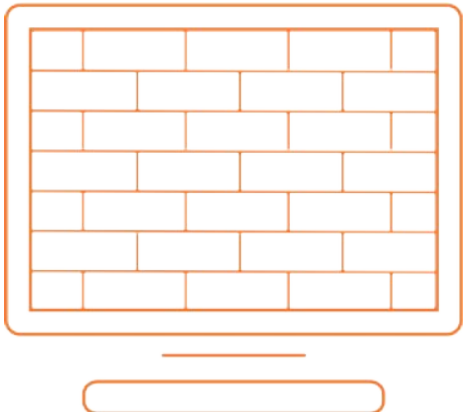
- Use a firewall to **rate limit UDP**, slowing the speed at which data can be transferred but not preventing it from happening
- **Analyze logs by hand** / using a visualization tool to find evidence of tunneling / data exfiltration after it's happened. Typically achieved by looking for abnormal query types (such as NULL), particular REGEXs and long queries all of the same length.
- **Tunnel endpoints can then be blocked** at the firewall level. Provides protection only if the attacker continues to use the same domain/server in the future.
- **Manually configure security appliances** (such as DDoS mitigation platforms) to block specific REGEXs, query types, long queries etc. Laborious process that risks blocking legitimate software such as anti-virus, that use DNS to check for new signatures, file hashes etc

Ineffective against APTs & software that can be configured to vary query length, query type etc. to avoid detection



DNS Firewall

- Like a firewall it implements a security policy (called RPZ – Response Policy Zone)
- Has triggers:
  - Queried domain name
  - IP address in response
  - IP address of querying client
- And actions:
  - Reject (NXdomain)
  - Block (no data)
  - Redirect
  - Passthrough and log



Detecting DNS Tunnels & Data Exfiltration

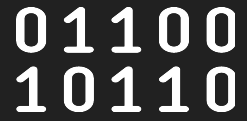
DNS Firewall approach (vendor based solutions)

Signature

Use **algorithms** to detect and block all known DNS Tunneling tools (in hardware, at line speed)

Reputation

Use **DNS Firewall** to automatically block known tunnel, C2 and data exfiltration endpoints and newly registered domains with Vendor backed database



Behavior

Use **DNS heuristics** to detect and block zero day data exfiltration, using real-time streaming analytics to mitigate against previously unseen code

DNS heuristics

- **Entropy**
  - Higher Entropy => more information transferred
  - Legitimate DNS names often have dictionary words or something that looks meaningful.
    Encoded names have a higher entropy. DNS names that have high entropy can be an indicator of tunneling.
- **Lexical**
  - Analysis of individual characters in domain names
  - non-letters (numbers or allowed special characters) character ratio
  - hex /A-F/ character ratio
  - vowel character ratio

DNS heuristics

- **N-Gram**
  - Detects non human like domain names based on character distribution.
    Focus is on 2- and 3-gram (i.e. sequences of 2 or 3 characters, or bigram and trigram analysis).

Source: https://books.google.com/ngrams

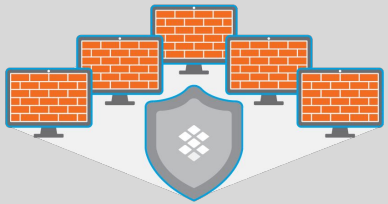
DNS heuristics

- **Gini index** – how often a randomly chosen character from the domain name would be incorrectly labeled if it was randomly labeled according to the distribution of characters in the domain name
- **Classification error** – measure of the diversity of characters in the string
- **Number of Labels** – number of domain labels in an FQDN payload
- **Frequency**: how often are requests being sent to the same recipient (typically multiple requests to same recipient are not common and indicate malicious activity)
  Are the queries being repeated at precise intervals?
- **Size** Higher payload size => more information transferred

1 Preventing communication over DNS

2 Stopping APTs/malware from using DNS

3 Defending DNS Infrastructure

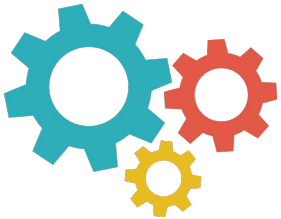


Type of DNS attacks

DNS reflection/DrDoS attacks	Using third-party DNS servers (mostly open resolvers) to propagate a DoS or DDoS attack
DNS amplification	Using a specially crafted query to create an amplified response to flood the victim with traffic
TCP/UDP/ICMP floods	Denial of service on layer 3 or 4 by bringing a network or service down by flooding it with large amounts of traffic
NXDomain attack	Attacks that flood DNS server with requests for non-existent domains, causing it to send NXDomain (non-existent domain) responses
Phantom domain attack	Attacks where a DNS resolver is forced to resolve multiple non-existent domains, causing it to consume resources while waiting for responses
DNS-based exploits	Attacks that exploit bugs or vulnerabilities in the DNS software
DNS cache poisoning	Corruption of DNS server cache data with a rogue domain or IP
Protocol anomalies	Causing the server to crash by sending malformed DNS packets and queries
Reconnaissance	Attempts by hackers to get information on the network environment before launching a DDoS or other type of attack via DNS queries or Zone Transfers
DNS tunneling	Tunneling of another protocol through DNS port 53 for malware insertion and/or data exfiltration
DNS hijacking	Modifying the DNS record settings to point to a rogue DNS server or domain

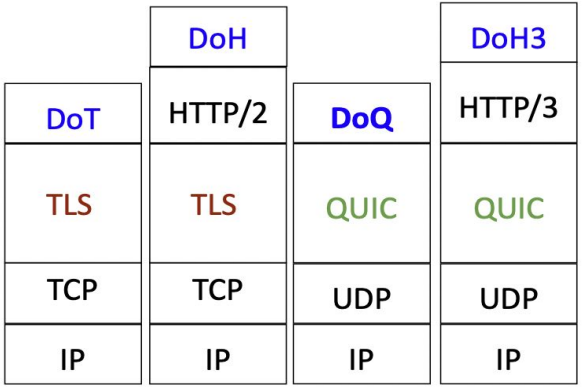
DNS-specific Exploits Volumetric/DDoS Attacks

Lab: DNS Reconnaissance



Practical Exercise: Reconnaissance of a domain

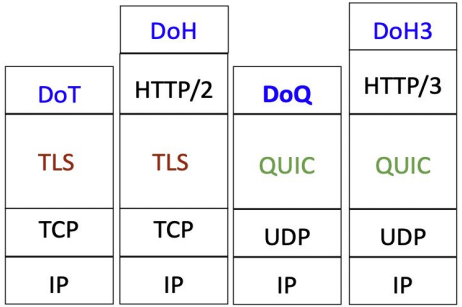
Improvements of DNS



- DNS **Transport** mechanisms
  - DNS over TLS (DoT): use DNS over TLS encryption on TCP port 853 (RFC 7858)
  - DNS over HTTPs (DoH/DoH3) : use HTTP2 or 3 for encrypting requests
  - DNS-over-QUIC (DoQ): use QUIC for data transport and encryption (UDP port 853 (also port used for DNS over DTLS)

- DNS **Data** authentication:
  - DNSSEC: use specific DNS records to ensure that the answer given is the authorized one

Comparing DoT/DoQ and DoH/DoH3



DoT	DoH
RFC 7858	RFC 8484
New Port 853	Port 443 (HTTPS)
Traffic can easily be blocked and not open normally by default in enterprise	Traffic looks like every other encrypted web traffic
No API capabilities	API capabilities
	Built-in browser support

DNSSEC

The original design of the Domain Name System (DNS) did not include security and allowed false DNS data to be returned.

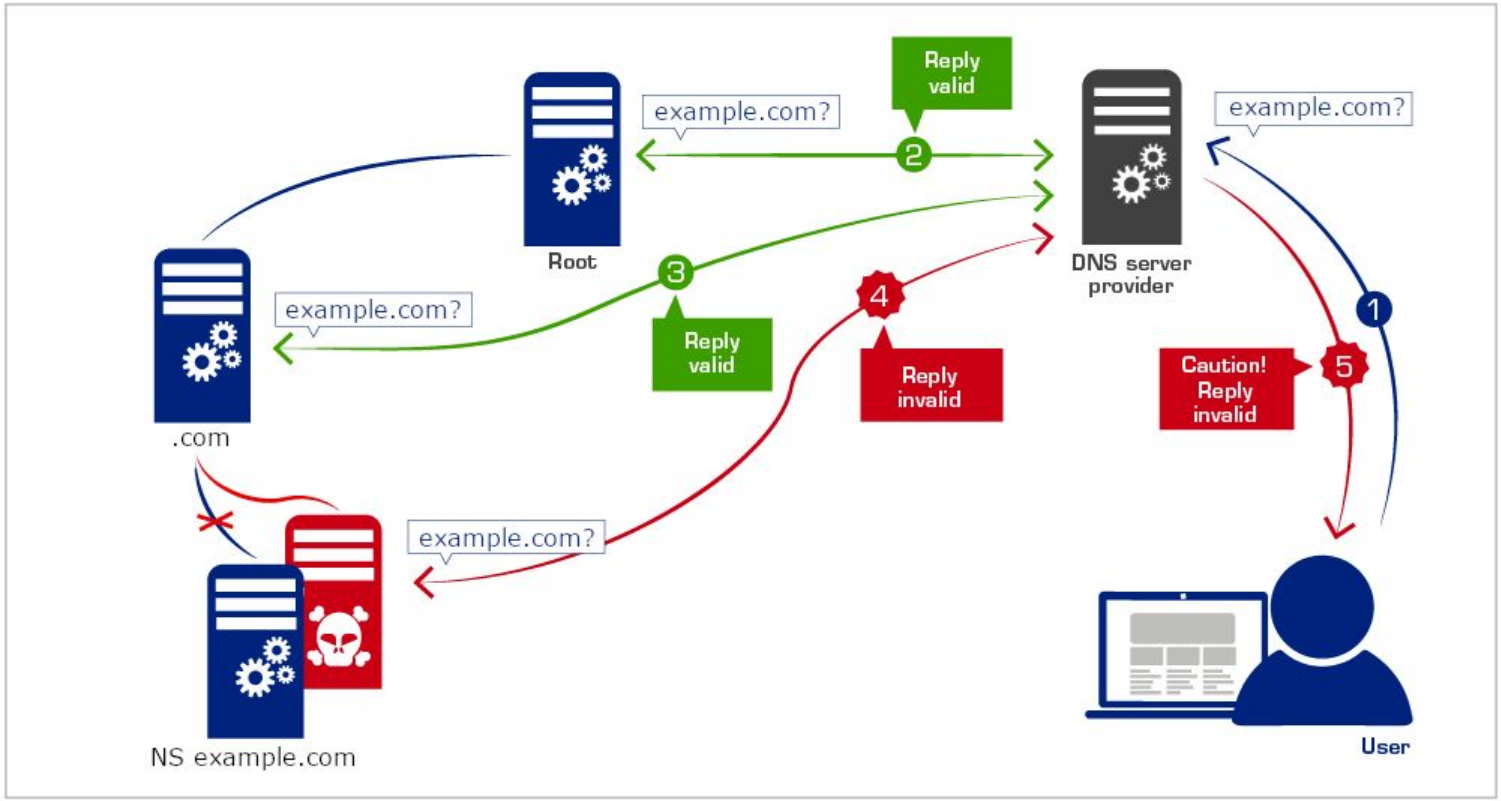
This required trust that the DNS answers were authentic and not modified.

Domain Name System Security Extensions (DNSSEC) is a set of extensions to DNS that provide DNS clients via a **digital signature** (resolvers) **origin authentication** of DNS data.

By validating the digital signature, a DNS resolver can check if the information is identical to the data published by the zone owner and served on an authoritative DNS server and thereby mitigate, such as ‘man-in-the-middle attacks’

DNSSEC doesn’t provide confidentiality of data, and the responses are **authenticated** but not **encrypted**.

DNSSEC


Root
example.com?
Reply valid
2
DNS server provider
example.com?
1
.com
example.com?
3
Reply valid
4
Reply invalid
Caution! Reply invalid
5
example.com?
NS example.com
User

Ressources

DNS RFCs

An Illustrated Guide to the Kaminsky DNS Vulnerability

DNS for Rocket Scientists


```
❯ dig -x 8.8.8.8 +short
dns.google.

❯ dig txt dns.google. +short
"https://xkcd.com/1361/"
"v=spf1 -all"
```