NETWORK SECURITY:

APPLICATION-LAYER & DNS

→ Application layer specifies the shared protocols used by hosts in a comm. network

→ The protocols include:

- DNS (Domain Name System)
- HTTP (Hypertext Transfer Protocol) / HTTPS
- SSL (Secure Socket Layer) / TLS (Transport Layer Security) — Secure, encrypted browsing
- IMAP (Internet Message Access Protocol) / POP (Post Office Protocol) / SMTP (Simple Mail Transfor Protocol) — Internet email protocols
- FTP (File Transfer Protocol) — Uploading & Downloading files
- SSH (Secure Shell) — Secure remote access protocol

URL

→ Uniform Resource Locator is a standardized format for describing the location and access method of resources via the internet

<scheme> :// <user> : <password> @ <host> : <port> / <url-path> ? <query-string>

<host> → <subdomain>. <domain>. <topdomain>

e.g. profile. facebook . com

DOMAINS

→ Domain name consists of 2/more labels separated by dots

e.g. inf.ed.ac.uk

TOP-LEVEL DOMAIN (TLD)
- gTLD: generic [.com, .org, .net]
- ccTLD: country-code [.uk, .ca]
- new TLDs [.scot, .tirol]

→ are managed by ICANN [Internet Corporation for Assigned Names & Numbers]

- keeps database of registered gTLDs

→ ccTLDs are managed by gov. organizations

DNS

→ The **domain name system** maps domain names to IP addresses
(The mapping is many-to-many)

→ DNS is a distributed database tht. stores resource records

- **Address (A) record**: IP address associated w/ a host name
- **Mail exchange (MX) record**: mail server of a domain
- **Name Server (NS) record**: authoritative server for a domain

NAME SERVER

→ keeps local database of DNS records

→ answers DNS queries

→ If record for particular domain is not in local dbase, can ask other name servers

NAME RESOLUTION

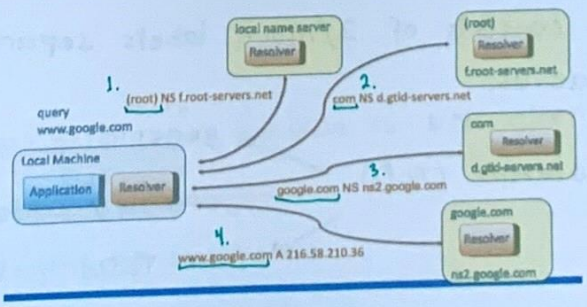
→ is the act of getting a domain name converted into its IP address

→ DNS Resolver: program tht. retrieves DNS records from a DNS server by connecting to a name server. Records received will be cached.

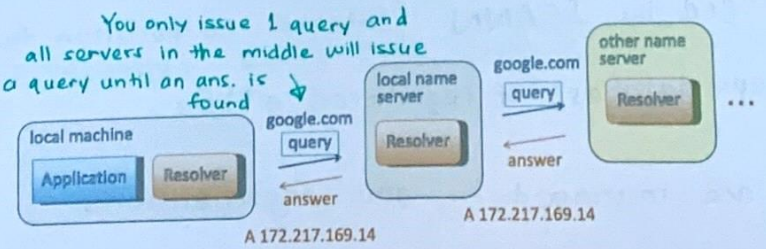
→ In an iterative resolution, name server refers client to authoritative server (e.g. a TLD server)

→ In a recursive resolution, name server queries another server, which may in turn query other servers on behalf of the requester.

ITERATIVE NAME RESOLUTION



RECURSIVE NAME RESOLUTION



CIRCULAR REFERENCE

→ caused when the authoritative name server for a domain is within the same domain

e.g. dns0.inf.ed.ac.uk is authoritative for inf.ed.ac.uk

→ To break this, we introduce GLUE RECORDS

- are of type A [IP Address]
- provide the IP address of a name server

e.g. inf.ed.ac.uk NS dns0.inf.ed.ac.uk

dns0.inf.ed.ac.uk A 129.215.160.240 → Glue Record!

DNS CACHING

→ DNS servers cache records that are results of queries for a specified amt of time

→ This prevents too much network traffic if a path in the DNS tree would be traversed for each query

↳ Root servers & TLD servers would be rapidly overloaded!

1) Resolver looks in cache for A record of query domain
2) Resolver looks in cache for NS record of longest suffix of query domain

→ Operating system maintains DNS cache

- Shared among all running apps.
- Can be displayed to all users

PRIVACY ISSUE!

- Browsing by other users can be monitored
- Incognito does not clear DNS cache

DNS CACHE POISONING

→ Corrupt DNS data is introduced into the DNS resolver's cache, causing the name server to return incorrect result record

→ This can happen bc. there is no authentication whether the response to the DNS query is received from the name server that it sent the request to

↳ Queries only have 16-bit request identifier in payload to match answers

→ Cache may be poisoned when:

a.) Query has predictable identifiers and return ports

b.) Attacker answers before authoritative name server

c.) Resolver ignores identifier & accepts unsolicited DNS records

HOW TO AVOID THIS ?

1) Query Randomization

→ Random request identifier (16 bits)
→ Random return port (16 bits)

So, prob. of guessing request ID & Port is:

$$\frac{1}{2^{16}} \times \frac{1}{2^{16}} = \boxed{\frac{1}{2^{32}}}$$

This is not enough

↳ Subdomain DNS Cache Poisoning

1) Attacker causes victim to send many DNS requests for nonexistent subdomains of target domain

2) Attacker sends victim forged NS responses for the requests (in hope to match the many requests the victim sent)
↳ include spoofed glue record pointing to the attacker's name server IP

2) DNSSEC

→ is a set of extensions to DNS which provide to DNS clients:

- authenticity of DNS answer origin
- authenticity of denial of existence
- integrity of reply

→ Uses public-key cryptography; signed DNS replies at each step