

Network Security: Application-Layer and Domain Name System

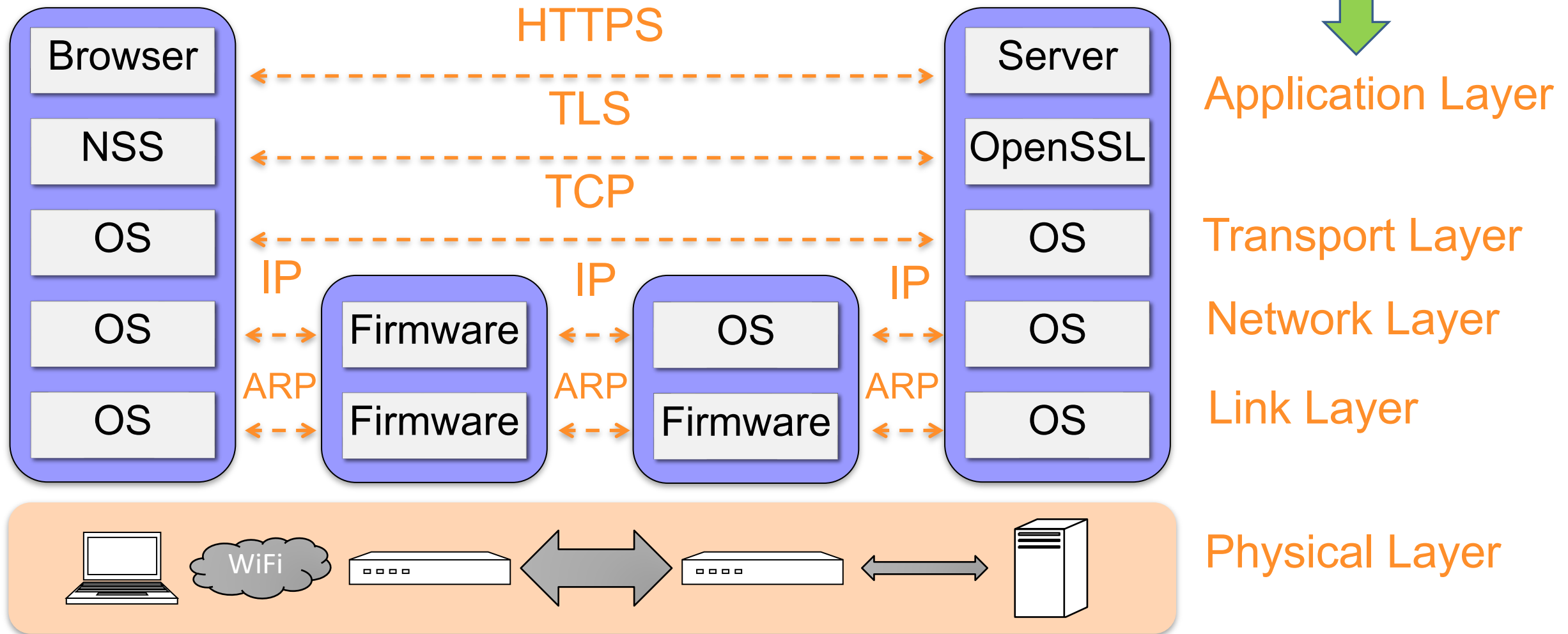
COMPUTER SECURITY

TARIQ ELAHI

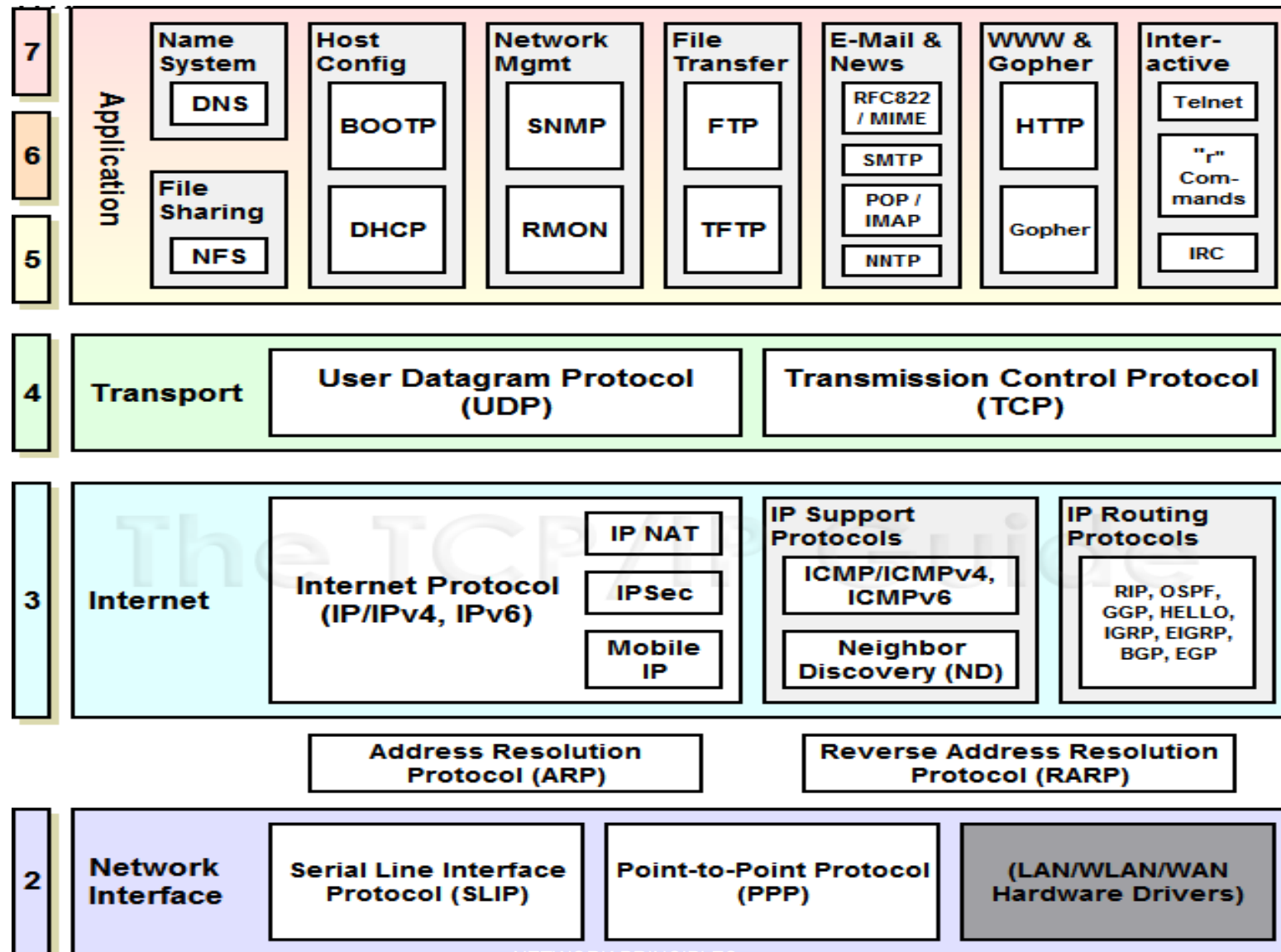
Some slides adapted from those by Markulf Kohlweiss, Myrto Arapinis, Kami Vania, and Roberto Tamassia



Internet Stack (simplified)



TCP/IP Model Mapped onto OSI



Sample Application-Layer Protocols

- Domain name system (**DNS**)
- Hypertext transfer protocol (**HTTP**)
- **SSL/TLS**. Protocol used for secure, encrypted browsing (**HTTPS**)
- **IMAP/POP/SMTP**. Internet email protocols
- File transfer protocol (**FTP**). An old but still used protocol for uploading and downloading files
- **Telnet**. Early remote access protocol
- **SSH**. More recent secure remote access protocol.



Other protocol examples [\[edit \]](#)

- [9P, Plan 9 from Bell Labs](#) distributed file system protocol
- [AFP, Apple Filing Protocol](#)
- [APPC, Advanced Program-to-Program Communication](#)
- [AMQP, Advanced Message Queuing Protocol](#)
- [Atom Publishing Protocol](#)
- [BEEP, Block Extensible Exchange Protocol](#)
- [Bitcoin](#)
- [BitTorrent](#)
- [CFDP, Coherent File Distribution Protocol](#)
- [CoAP, Constrained Application Protocol](#)
- [DDS, Data Distribution Service](#)
- [DeviceNet](#)
- [eDonkey](#)
- [ENRP, Endpoint Handlespace Redundancy Protocol](#)
- [FastTrack \(KaZaa, Grokster, iMesh\)](#)
- [Finger, User Information Protocol](#)
- [Freenet](#)
- [FTAM, File Transfer Access and Management](#)
- [Gopher, Gopher protocol](#)
- [HL7, Health Level Seven](#)
- [HTTP, HyperText Transfer Protocol](#)
- [H.323, Packet-Based Multimedia Communications System](#)
- [IMAP, Internet Message Access Protocol](#)
- [IRCP, Internet Relay Chat Protocol](#)
- [IPFS, InterPlanetary File System](#)
- [Kademlia](#)
- [LDAP, Lightweight Directory Access Protocol](#)
- [LPD, Line Printer Daemon Protocol](#)
- [MIME \(S-MIME\), Multipurpose Internet Mail Extensions and Secure MIME](#)
- [Modbus](#)
- [MQTT Protocol](#)
- [Netconf](#)
- [NFS, Network File System](#)
- [NIS, Network Information Service](#)
- [NNTP, Network News Transfer Protocol](#)
- [NTCIP, National Transportation Communications for Intelligent Transportation System Protocol](#)
- [NTP, Network Time Protocol](#)
- [OSCAR, AOL Instant Messenger Protocol](#)
- [POP, Post Office Protocol](#)
- [PNRP, Peer Name Resolution Protocol](#)
- [RDP, Remote Desktop Protocol](#)
- [RELP, Reliable Event Logging Protocol](#)
- [Rlogin, Remote Login in UNIX Systems](#)
- [RPC, Remote Procedure Call](#)
- [RTMP, Real Time Messaging Protocol](#)
- [RTP, Real-time Transport Protocol](#)
- [RTPS, Real Time Publish Subscribe](#)
- [RTSP, Real Time Streaming Protocol](#)
- [SAP, Session Announcement Protocol](#)
- [SDP, Session Description Protocol](#)
- [SIP, Session Initiation Protocol](#)
- [SLP, Service Location Protocol](#)
- [SMB, Server Message Block](#)
- [SMTP, Simple Mail Transfer Protocol](#)
- [SNTP, Simple Network Time Protocol](#)
- [SSH, Secure Shell](#)
- [SSMS, Secure SMS Messaging Protocol](#)
- [TCAP, Transaction Capabilities Application Part](#)
- [TDS, Tabular Data Stream](#)
- [Tor \(anonymity network\)](#)
- [Tox](#)
- [TSP, Time Stamp Protocol](#)
- [VTP, Virtual Terminal Protocol](#)
- [Whois \(and RWhois\), Remote Directory Access Protocol](#)
- [WebDAV](#)
- [X.400, Message Handling Service Protocol](#)
- [X.500, Directory Access Protocol \(DAP\)](#)
- [XMPP, Extensible Messaging and Presence Protocol](#)



What is a URL?

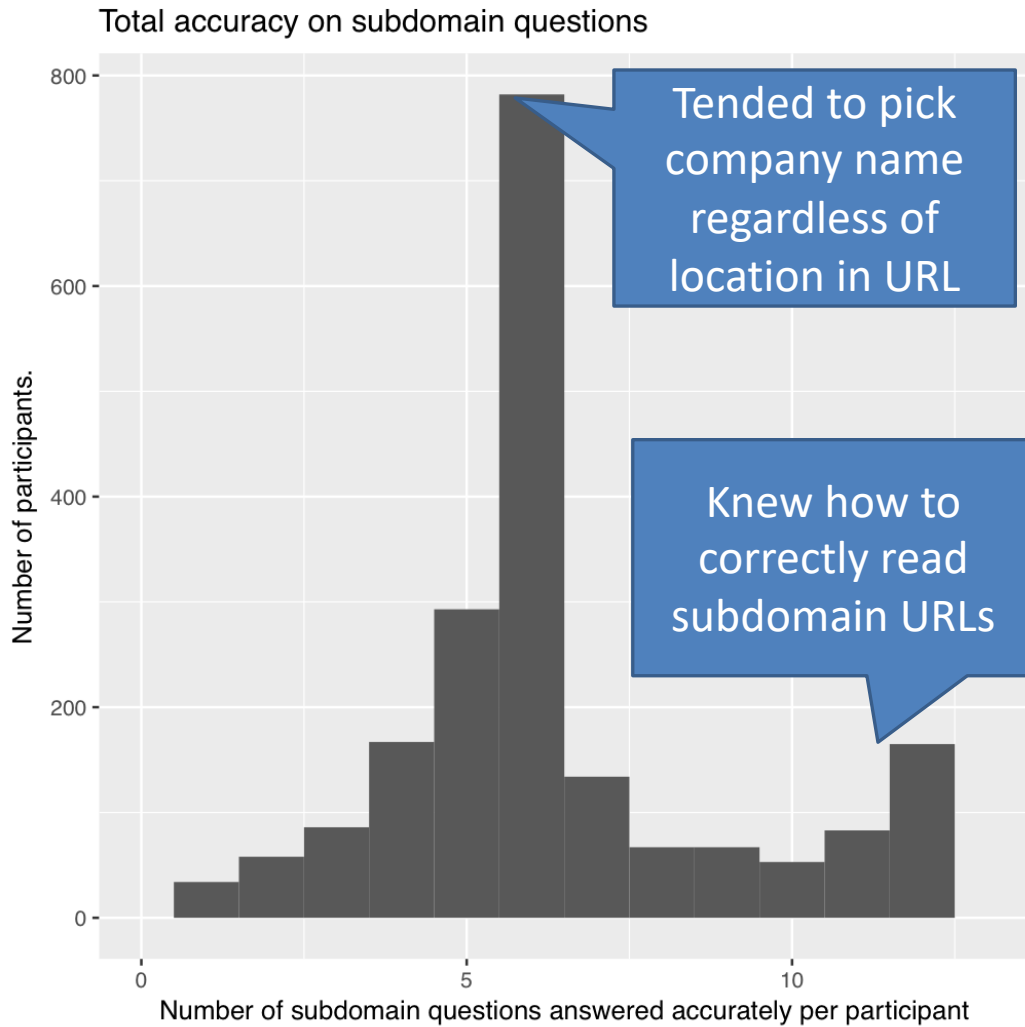
- Uniform Resource Locators (URLs) are a standardized format for describing the location and access method of resources via the internet.

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

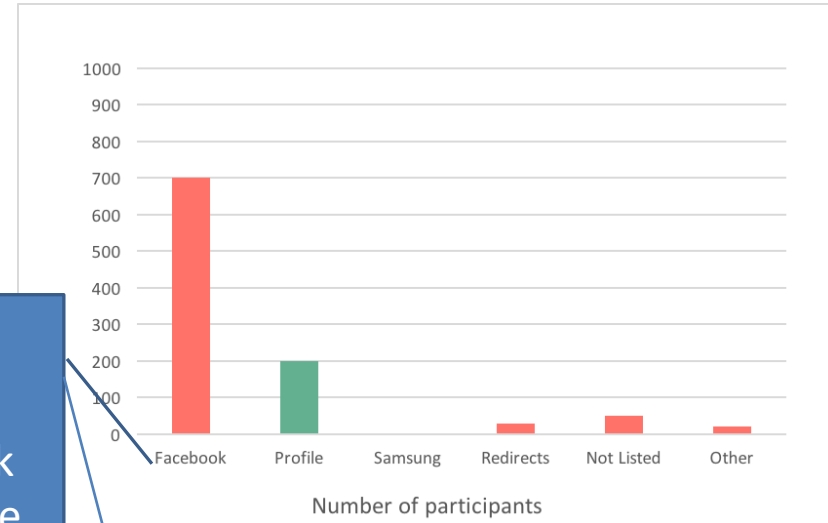
`<subdomain>.<domain>.<topdomain>`

eg. `https://profile.facebook.com`

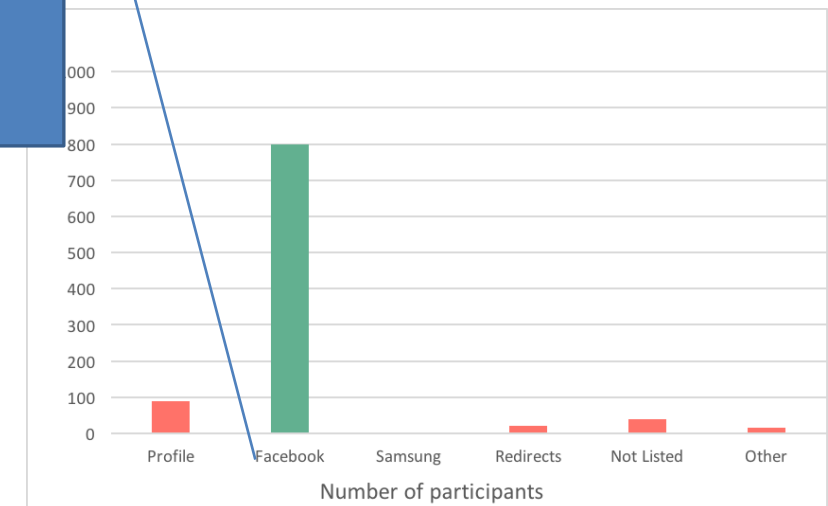


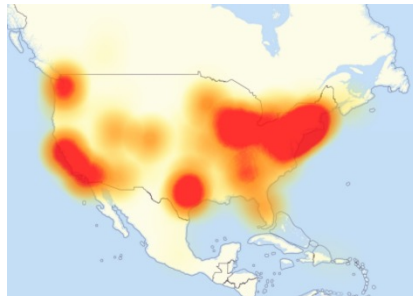


<https://facebook.profile.com>



<https://profile.facebook.com>





DNS Servers
are soft
targets for
attackers, take
out the
mapping and
the website
goes “offline”

DoS attack on major DNS provider brings Internet to morning crawl [Updated]

Dyn's US East region hit hardest in attack that affected Twitter, Reddit.

SEAN GALLAGHER - OCT 21, 2016 1:59 PM UTC

118



Update (12:04p ET): A second wave of DDoS attacks against Dyn is underway, as of noon Eastern Time today. Dyn is continuing to work on the issue. Our original story follows below; further updates will be added as information becomes available.

A distributed denial of service attack against Dyn, the dynamic DNS service, affected the availability of dozens of major websites and Internet services this morning, including Twitter and Reddit. The attack, **which began this morning at 7:10am Eastern Time** (12:10pm UK), is apparently focused on Dyn's US East Coast name servers.

"This morning, Dyn received a global DDoS attack on our Managed DNS infrastructure in the east coast of the United States," Doug Madory, Director of Internet Analysis at Dyn, said in an e-mail sent to Ars this morning. "DNS traffic resolved from east coast name server locations are experiencing a service interruption during this time." By 9:20am ET this morning, Dyn had mitigated the attack and services returned to normal.

[Update, 1:20 PM ET] Less than three hours later, the attack began again, and is still in progress.



Syrian group cited as New York Times outage continues

By Heather Kelly, CNN

🕒 Updated 1330 GMT (2130 HKT) August 29, 2013



The N
Stor
New
probl
Syria
York
The hackers gained access to a Melbourne IT reseller account using a phishing email and proceeded to change the DNS records of multiple domains, including NYTimes.com, according to the company.

The group is loyal to Syrian President Bashar Al-Assad

Twitter also experienced problems on Tuesday due to a similar attack

multiple attacks on media websites in recent months and, on Twitter, took credit for a sophisticated hack that had hobbled the Times' news site for roughly 20 hours.

"The @nytimes attack was going to deliver an anti-war message but our server couldn't last for 3 minutes," the group posted [on its Twitter](#)

[feed](#) at about 9:40 Wednesday morning.



Domain Name System

The **domain name system** (DNS) is an application-layer protocol

Basic function of DNS

Map domain names to IP addresses

The mapping is many to many

Examples:

www.ed.ac.uk and **edwc.is.ed.ac.uk**
map to 129.215.228.101

google.com maps to 216.58.213.110,
198.7.237.249, and other addresses

More generally, DNS is a distributed database that stores **resource records**

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



Domains

Domain name

- Two or more labels, separated by dots (e.g., [inf.ed.ac.uk](#))

Top-level domain (TLD)

- Generic (gTLD), e.g., [.com](#), [.org](#), [.net](#)
- Country-code (ccTLD), e.g., [.ca](#), [.it](#)
- New top level domains, e.g., [.scot](#), [.tirol](#)

ICANN

- (non-profit) Internet Corporation for Assigned Names and Numbers
- Keeps database of registered gTLDs ([InterNIC](#))
- Accredits registrars for gTLDs

gTLDs

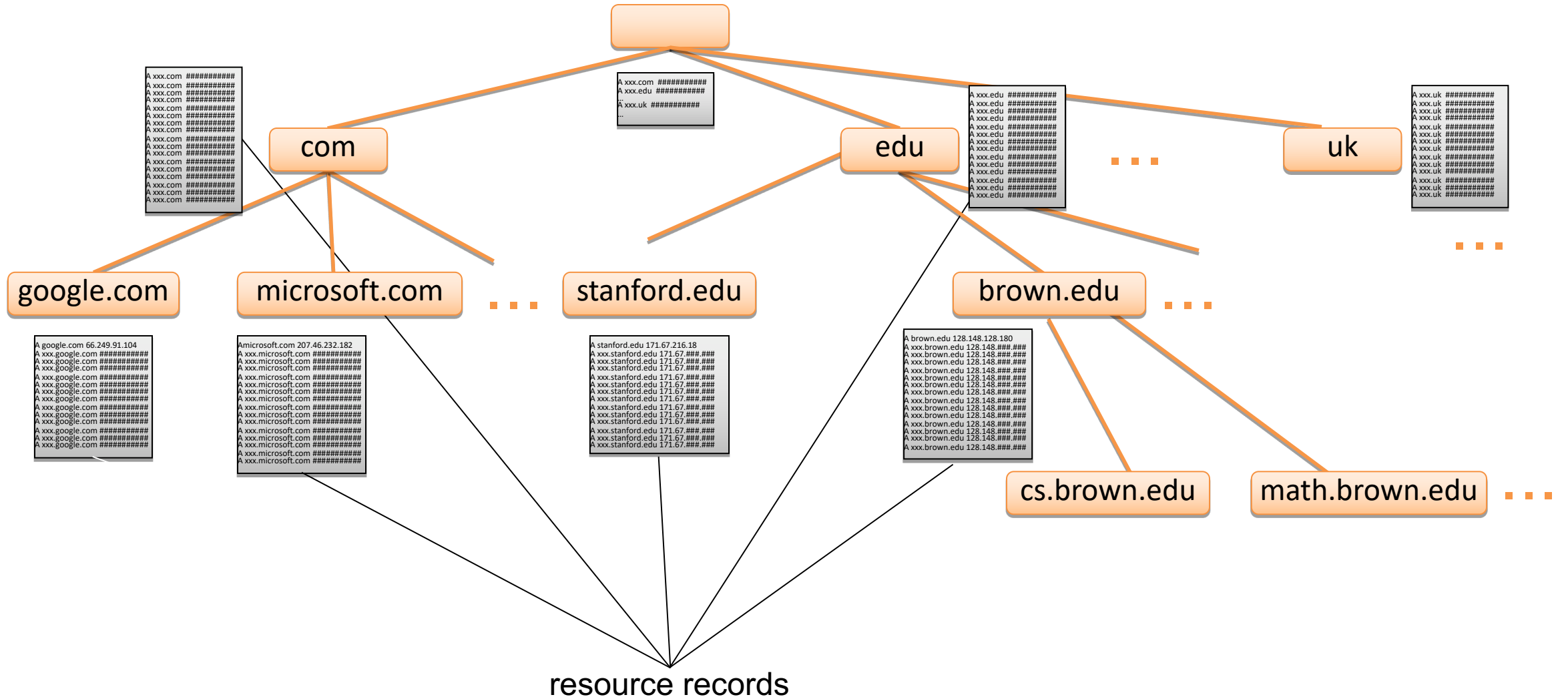
- Managed by ICANN

ccTLDs

- Managed by government organizations



DNS Tree



Name Servers

- Name server
 - Keeps local database of DNS records
 - Answers DNS queries
 - Can ask other name servers if record not in local database
- Authoritative name server
 - Stores reference version of DNS records for a zone (partial tree)
- Examples
 - [dns0.ed.ac.uk](#) is authoritative for [ed.ac.uk](#) and [dns0.inf.ed.ac.uk](#) for [inf.ed.ac.uk](#)
- Root servers
 - Authoritative for the root zone (TLDs)
 - [\[a-m\].root-servers.net](#)
 - Supervised by ICANN



Name Resolution

- Resolver
 - Program that retrieves DNS records
 - Connects to a name server (default, root, or given)
 - E.g., **dig** in Linux and **nslookup** in Windows
 - Caches records received

Iterative resolution

Name server refers client to authoritative server (e.g., a TLD server) via an NS record

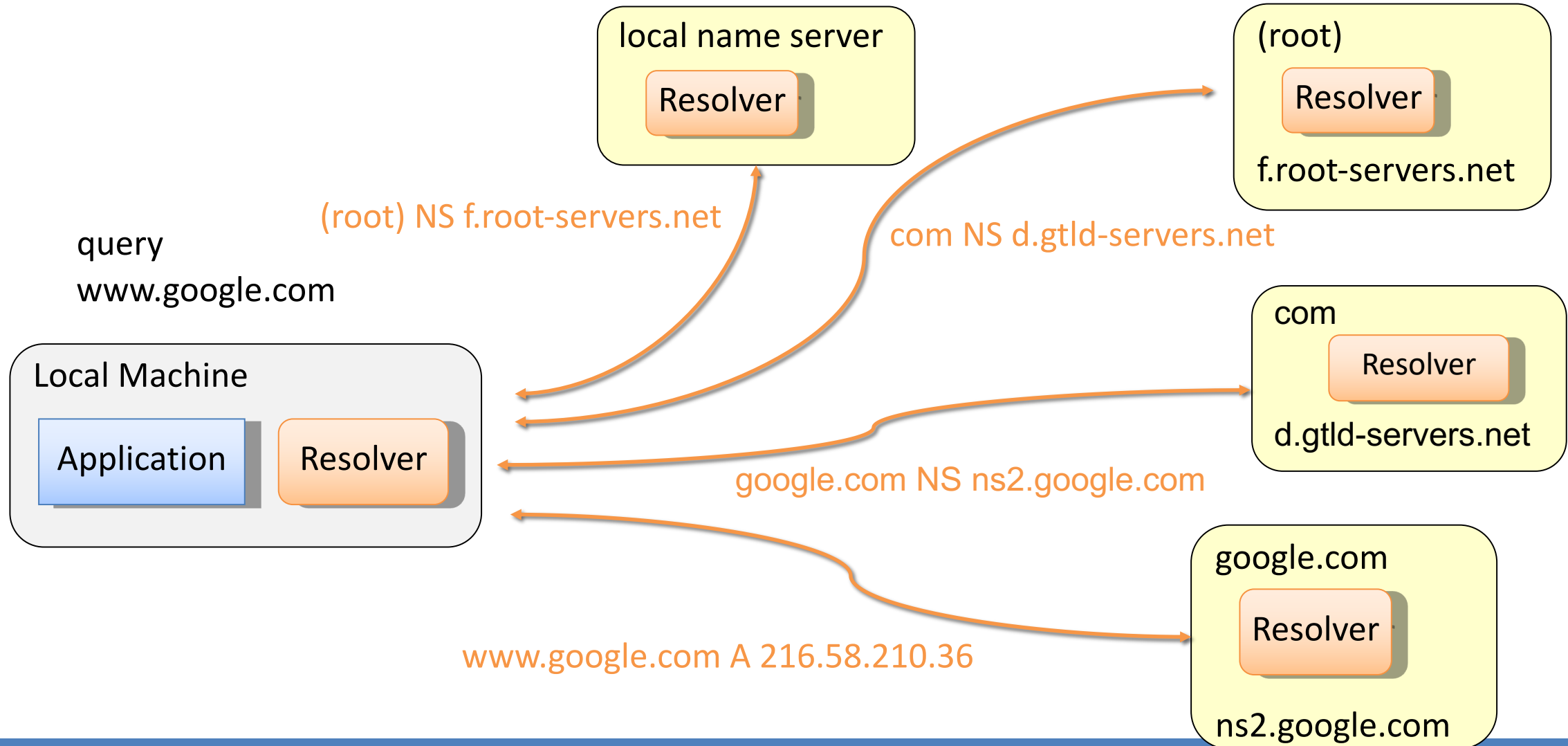
Repeat

Recursive resolution

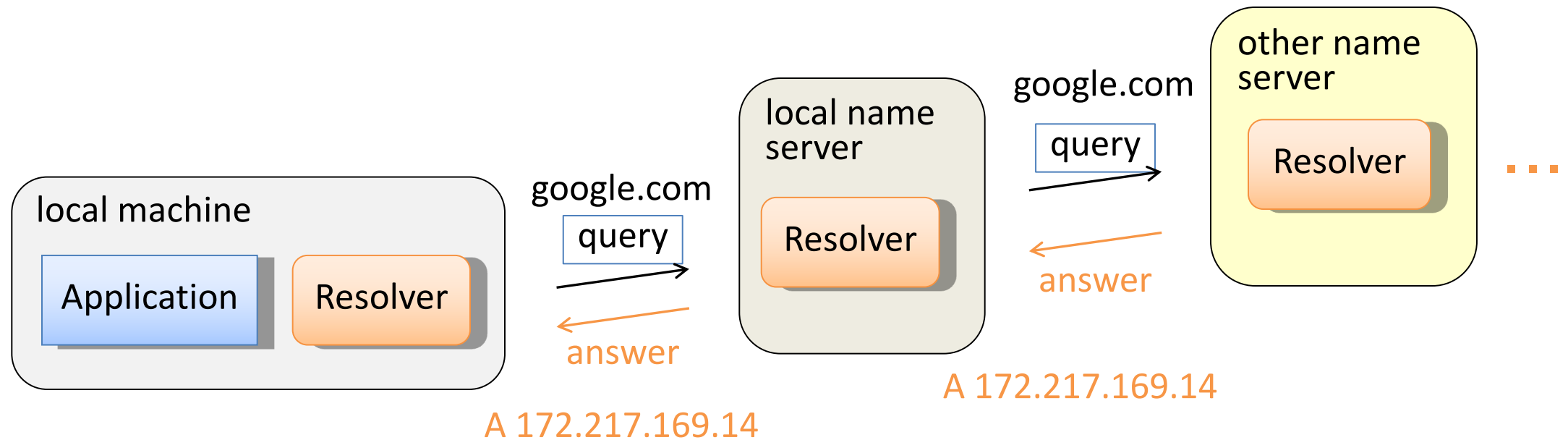
Name server queries another server and forwards the final answer (e.g., A record) to client



Iterative Name Resolution



Recursive Name Resolution



DNS Caching

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

Root servers and TLD servers would be rapidly overloaded

DNS servers **cache** records that are results of queries for a specified amount of time

Time-to-live field

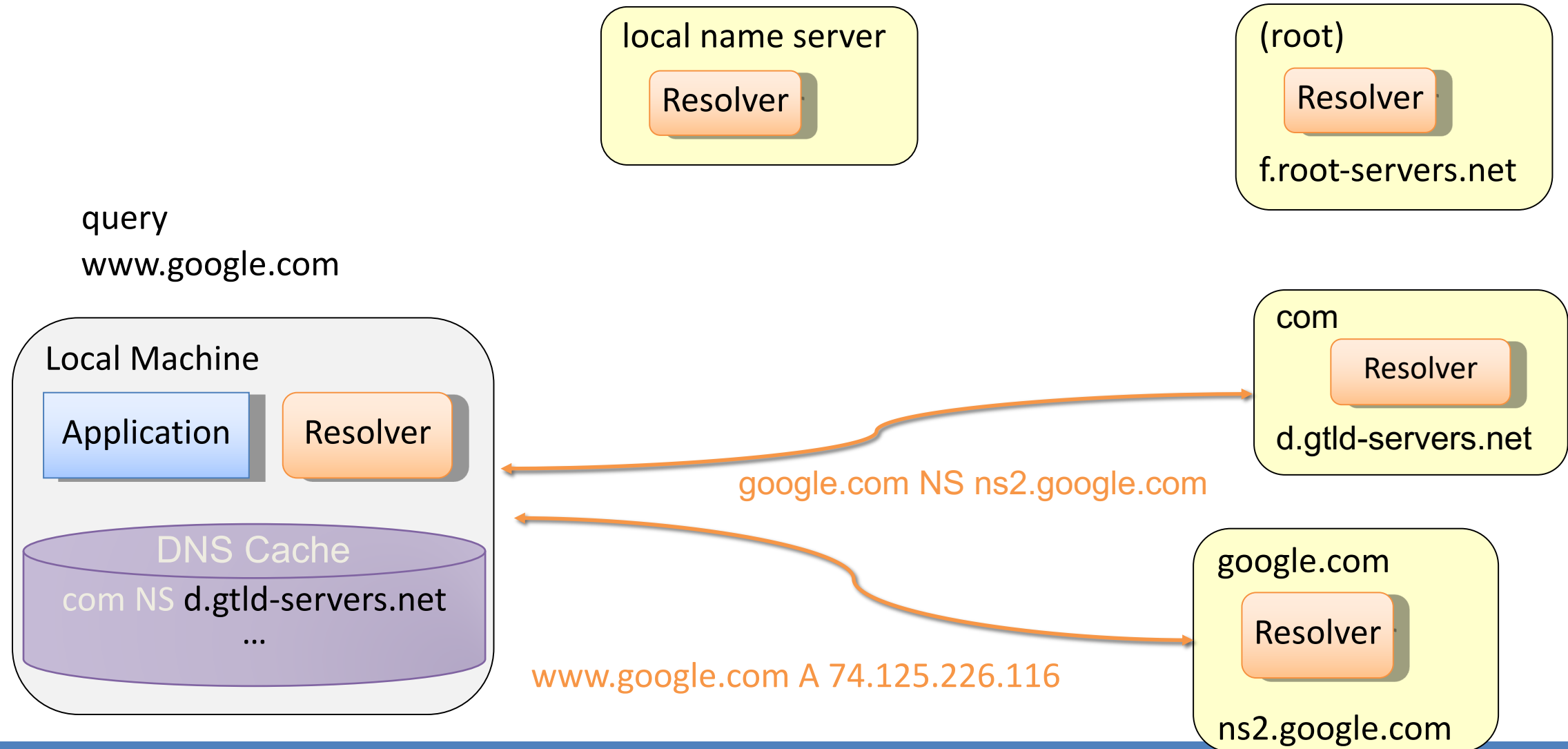
DNS queries with caching

First, resolver looks in cache for A record of query domain

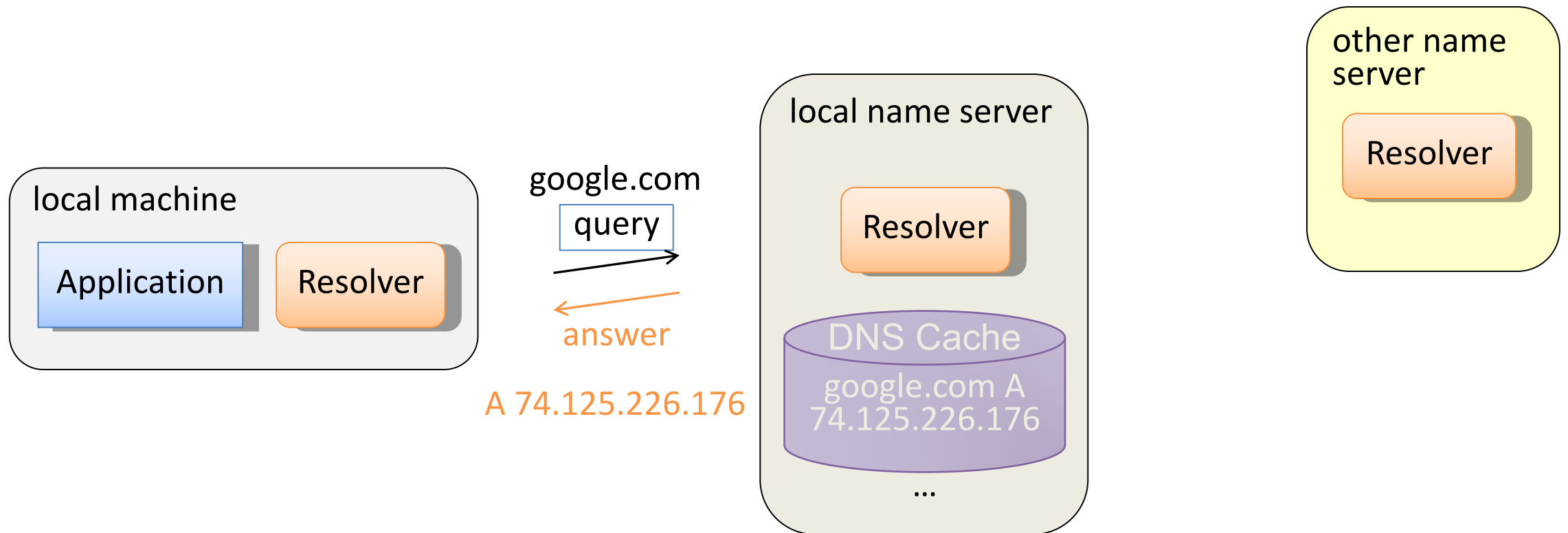
Next , resolver looks in cache for NS record of longest suffix of query domain



Iterative Name Resolution with Caching



Recursive Name Resolution with Caching



Local DNS Cache

Operating system maintains DNS cache

Shared among all running applications

Can be displayed to all users

View DNS cache in Windows with command `ipconfig /displaydns`

Clear DNS cache in Windows with command `ipconfig /flushdns`

Privacy issues

Browsing by other users can be monitored

Note that private/incognito browsing does not clear DNS cache

```
C:\Users\marku>ipconfig /displaydns
```

```
Windows IP Configuration
```

```
arstechnica.com
```

```
-----
```

```
Record Name . . . . . : arstechnica.com
```

```
Record Type . . . . . : 1
```

```
Time To Live . . . . . : 128
```

```
Data Length . . . . . : 4
```

```
Section . . . . . : Answer
```

```
A (Host) Record . . . : 50.31.169.131
```



DNS Cache Poisoning

Basic idea

Give a DNS server a false address record and get it cached

DNS query mechanism

Queries issued over UDP on port 53

16-bit **request identifier** in payload to match answers with queries

No authentication

Cache may be poisoned when a resolver

Query has predictable identifiers and return ports

Attacker answers before authoritative name server

Ignore identifier, accepts unsolicited DNS records

Early versions of BIND (popular DNS software) vulnerable to cache poisoning



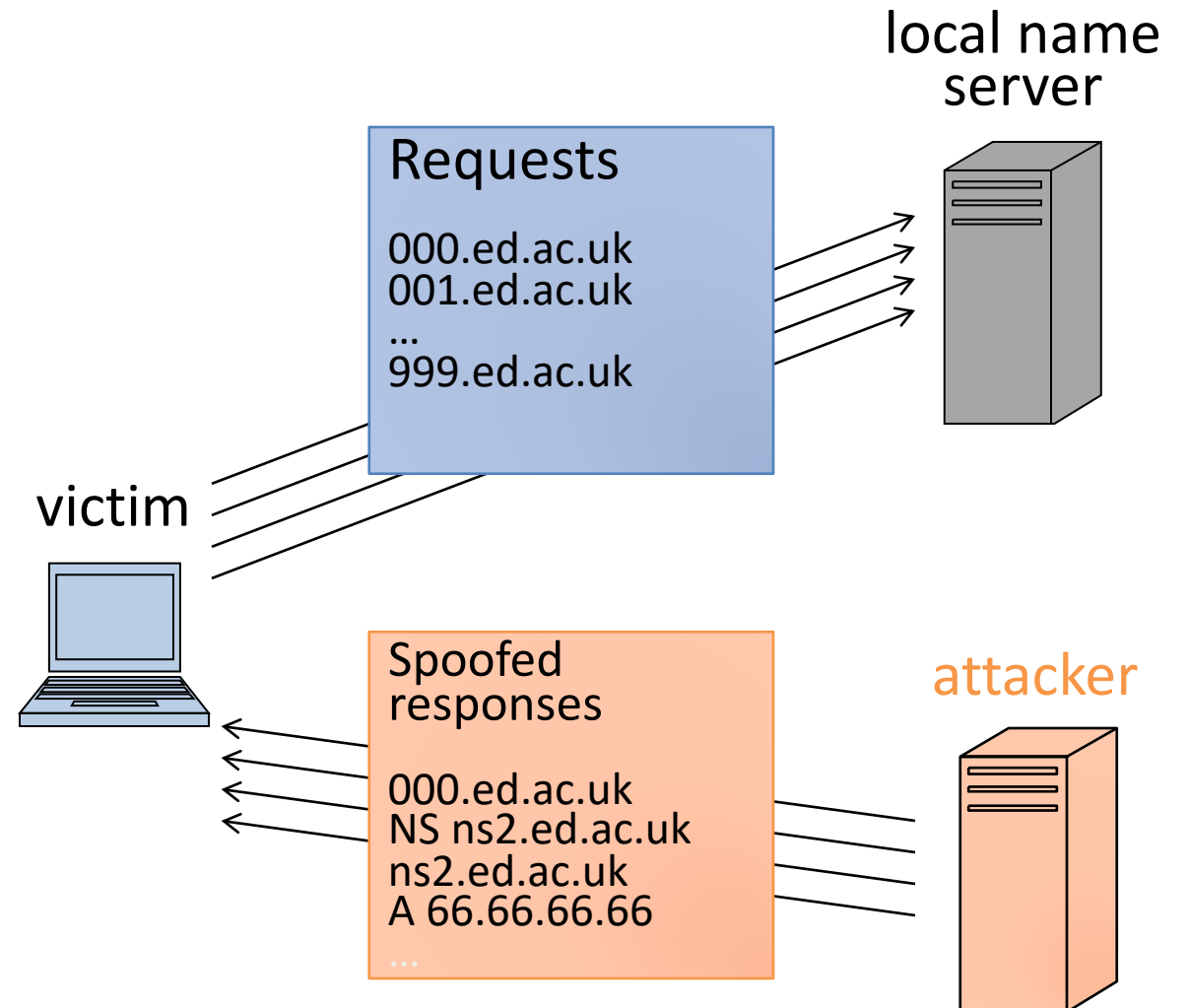
DNS Cache Poisoning Defenses

- Query randomization
 - Random request identifier (16 bits)
 - Random return port (16 bits)
 - Probability of guessing request ID **or** return port
 - $1 / 2^{16} = 0.0015\%$
 - Probability of guessing request ID **and** return port is
 - $1 / 2^{32}$ (less than one in four billion)
- Single guess low likelihood
 - But many guesses?



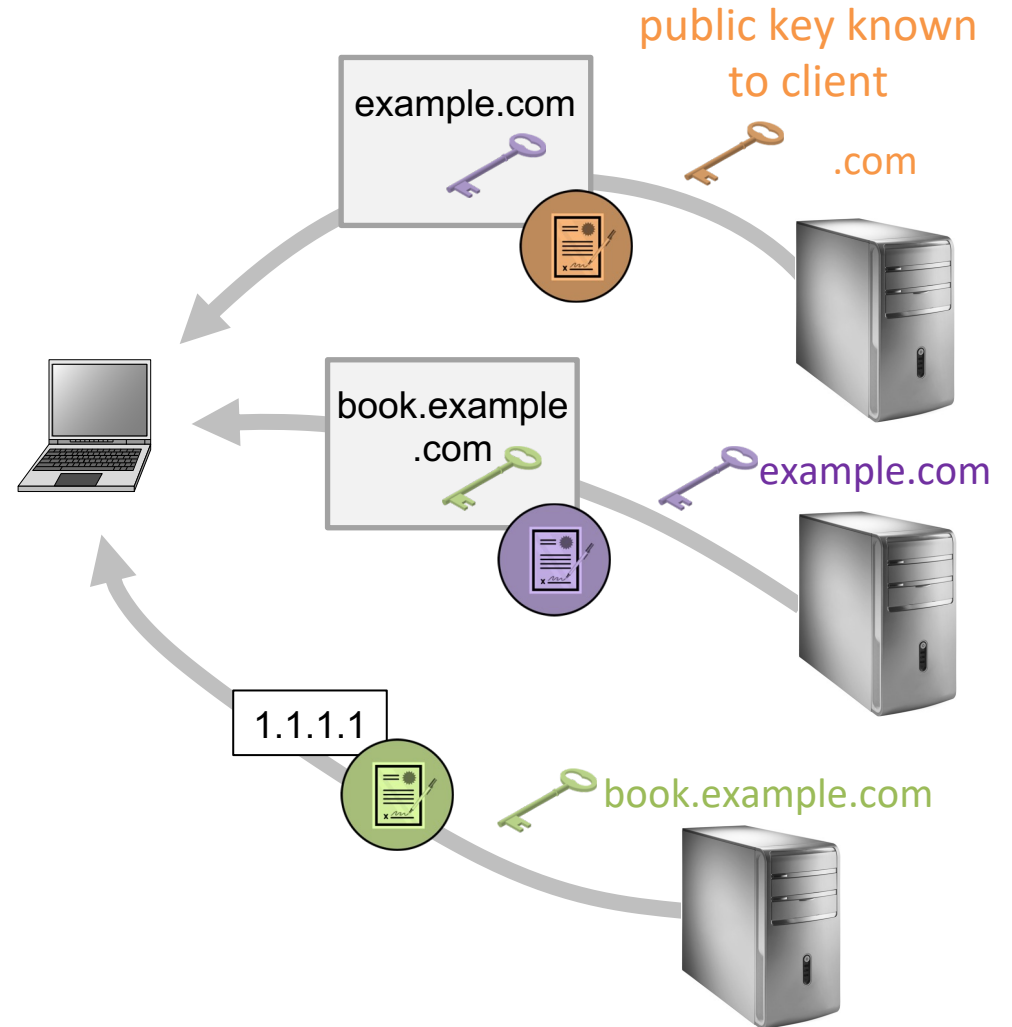
Subdomain DNS Cache Poisoning (Kaminsky)

- Attacker causes victim to send
 - Many DNS requests for nonexistent subdomains of target domain
- Attacker sends victim
 - Forged NS responses for the requests
- Format of forged response
 - Random ID
 - Correct NS record
 - Spoofed glue record pointing to the attacker's name server IP



DNSSEC

- Goals
 - Authenticity of DNS answer origin
 - Integrity of reply
 - Authenticity of denial of existence
- Implementation
 - Signed DNS replies at each step
 - Public-key cryptography
- Slow deployment
 - Root servers support since 2010



What We Have Learned

- How DNS operates
 - Distributed database
 - Resolvers and name servers
 - Iterative vs. recursive resolution
 - Caching
- DNS cache poisoning attacks and early defenses
- DNSSEC

