Evolving Landscape of the Domain Name System (DNS)

Shumon Huque

January 20th 2023

ECE Seminar Series

University of Virginia, Charlottesville, VA

"The Evolving Landscape of the Domain Name System (DNS)"
Shumon Huque

This talk will give an overview of recent developments in the evolution of the Domain Name System (DNS), the distributed global database that provides name to address mappings (and more) for the Internet. After a brief review of the DNS, it will cover how the worldwide DNS ecosystem has been evolving in recent years, and where it might be going in the future. Topics will include DNSSEC (cryptographic authentication of DNS data), DANE (DNSSEC as a PKI for applications), DNS Privacy (DNS over authenticated and encrypted transports such as TLS, HTTPS, QUIC), impacts of middleboxes, industry consolidation trends, and tensions between the deployment of new DNS features and the prevailing security postures of corporate networks. Lastly, we'll discuss how academic researchers could more effectively participate in the engineering and evolution of the DNS system.

[Slides: https://www.huque.com/talks/2023-01-evolving-dns.pdf]

Standard disclaimer

This talk contains *some* of my own views on a range of topics, and not those of any company that currently employs me, or has employed me in the past.

Who am I?

- A technologist with Salesforce, a cloud computing company
  - Software Engineering Architect
  - Product Owner of central DNS services
- Previously
  - Principal Scientist at Verisign Labs
  - Misc roles at the University of Pennsylvania (Systems Programmer, Network Engineer, Engineering Director, Adjunct Faculty)
- Educational background
  - Bachelors and Masters degrees in Computer Science from the University of Pennsylvania

Rough Outline of this talk

- DNS early history and protocol overview
- Recent changes in the DNS industry
- DNS Protocol evolution
  - Standard vs non-standard features
  - DNSSEC & DANE
  - DNS Privacy
  - HTTPS and SVCB (time permitting)
- Research areas and how academia can help

Domain Name System review

DNS: “Domain Name System”

- “**Domain Names**” are a way to identify Internet resources (e.g. a computer, network, or application service) in a human friendly textual form. e.g. **www.amazon.com**
- Old! Current base protocol RFC 1034, 1035, ~ 1987.
  - Hierarchical, tree-structured namespace composed of domain names.
  - Globally distributed database, with decentralized administration.
  - Client server lookup protocol.
- “Development of the Domain Name System” - P. Mockapetris, SIGCOMM88

IP Address Lookup (“A” record)

```
www.google.com.    300   IN   A   172.217.15.100
```

Human friendly “domain name”

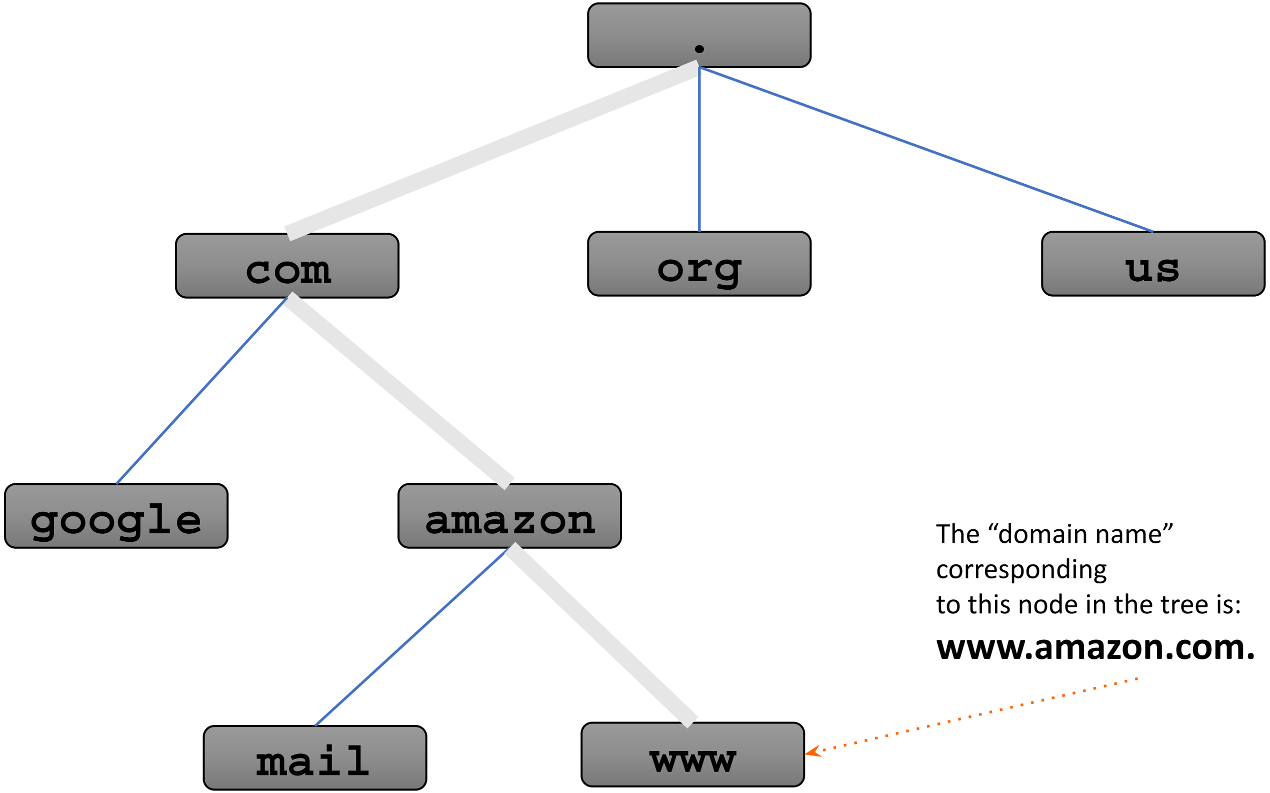
Numeric IP address, by which computers communicate

There are many more record types defined … a few examples follow

Record Type	Usage/Description
A	IPv4 Address record
AAAA	IPv6 Address record
SOA	Zone ‘Start of Authority’ parameters
NS	Name Server - zone apex or child zone delegations
MX	Mail Exchanger
PTR	Pointer (most commonly used for Reverse DNS)
TXT	Free form text (many apps encode application specific semantics)
SRV	Service Location record
NAPTR	Naming Authority Pointer record

Record Type	Usage/Description
DNAME	Domain Subtree Redirection
DNSKEY	DNSSEC Public Key
DS	Delegation Signer
NSEC	Authenticated Denial of Existence
NSEC3	Authenticated Denial of Existence (newer version)
CAA	Certification Authority Authorization
<many more>	
<metatypes>	

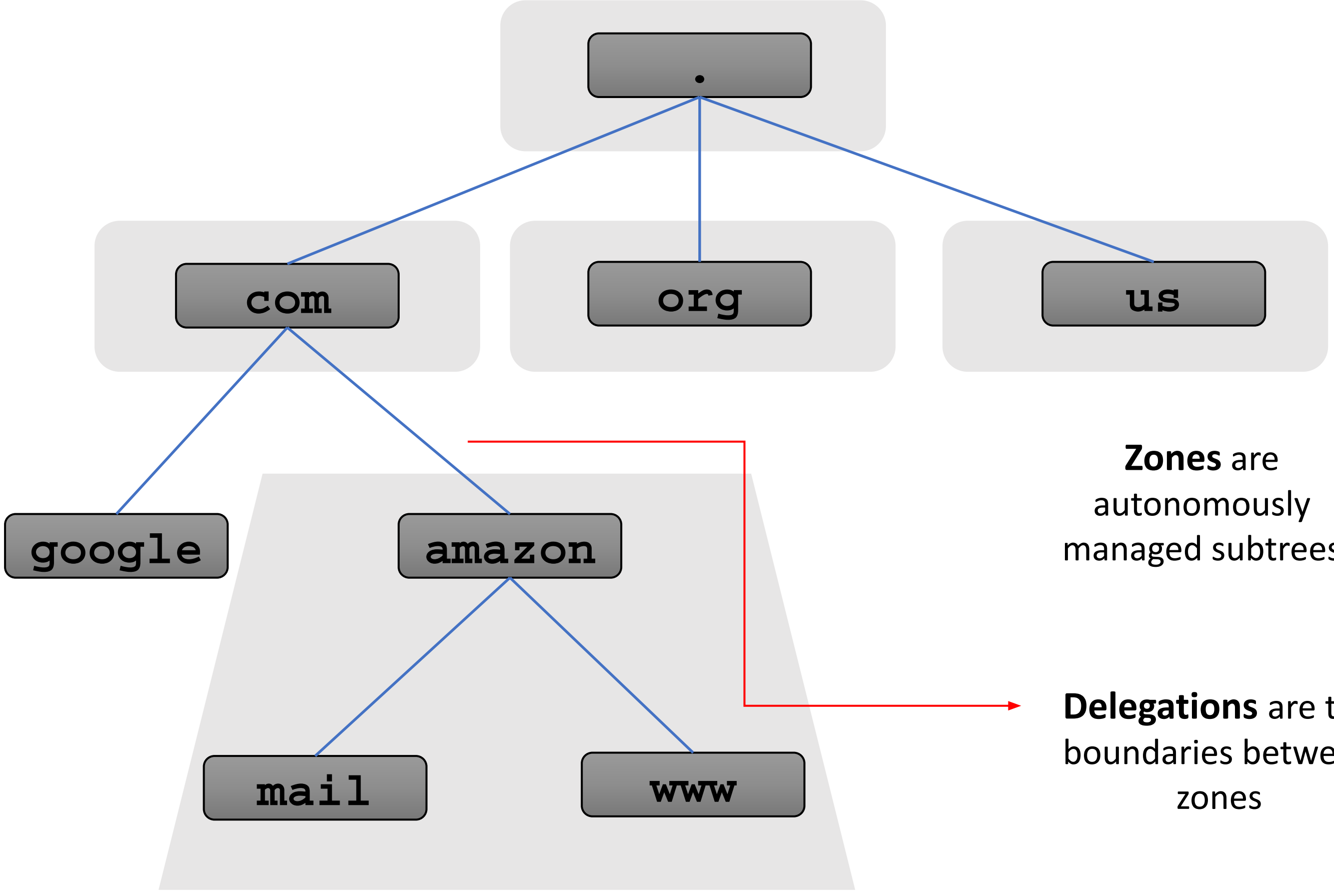
Full list at: https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4



The “domain name” corresponding
to this node in the tree is:

www.amazon.com.

Hierarchical Structure

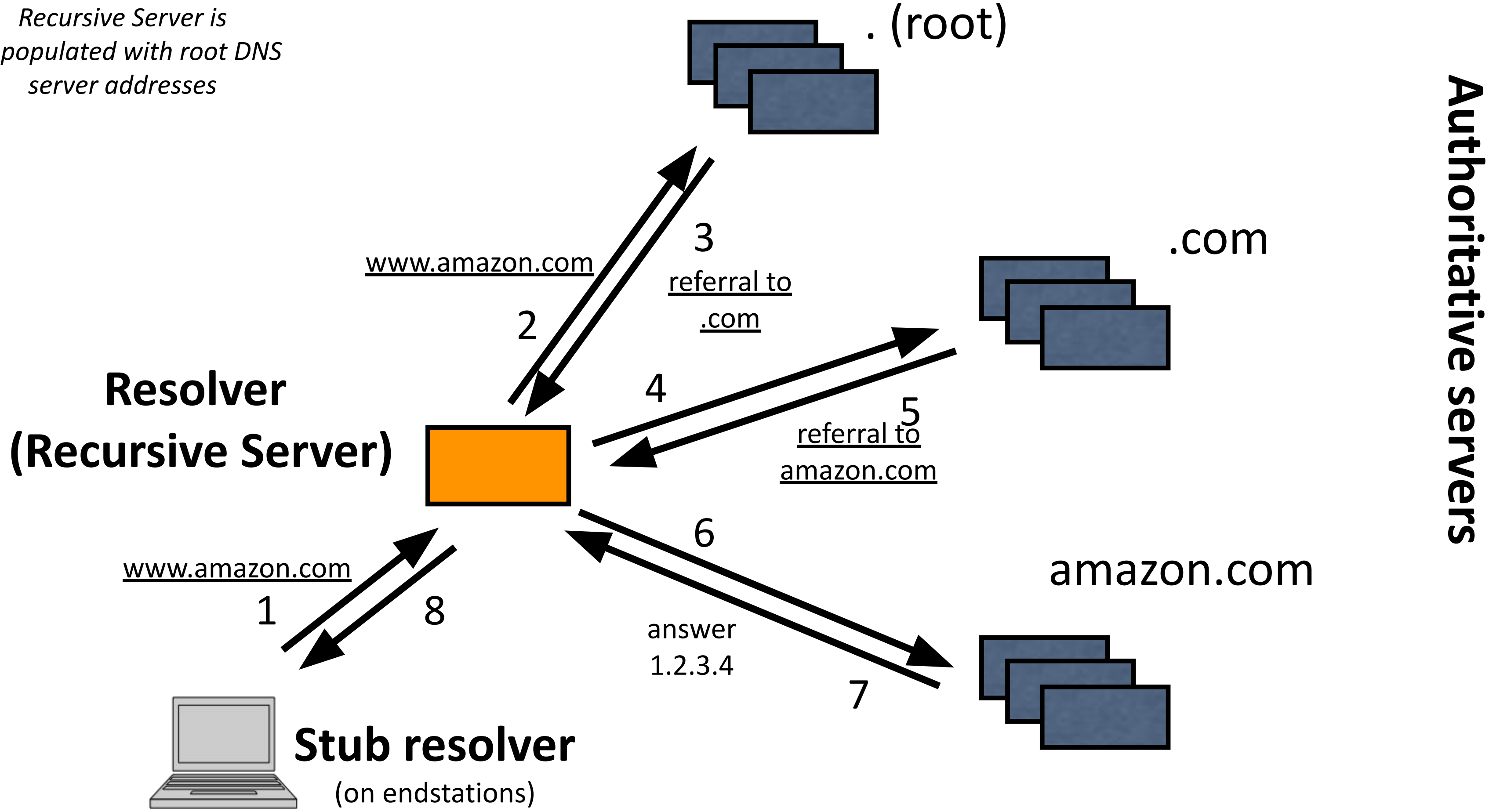


Zones are autonomously managed subtrees

Delegations are the boundaries between zones

Hierarchical Structure, but Decentralized Administration

Recursive Server is prepopulated with root DNS server addresses



. (root)

.com

amazon.com

Authoritative servers

Resolver (Recursive Server)

Stub resolver

(on endstations)

1 www.amazon.com

2 www.amazon.com

3 referral to .com

4

5 referral to amazon.com

6

7 answer 1.2.3.4

8

DNS Industry recent changes

Recent DNS industry changes of note

- IANA Functions Transition
  - Change in operation of the DNS Root
- Expansion of the Top Level Domains
  - New gTLD program

Root Zone & IANA Functions Transition

- Until recently, 3 partners were involved in **operation of the Root**
  - **US Government** (NTIA, Dept of Commerce) - Authorizer
  - **ICANN** (Internet Corp for Assigned Names & Numbers) - Root Zone Manager
  - **Verisign** (Private enterprise) - Root Zone Maintainer
  - (also 12 root zone "operator" organizations)

- IANA Functions Transition (mid 2013 - late 2016)
  - **Withdrawal of US government oversight and involvement**
  - Structural changes to ICANN's governance: "Multi-stakeholder model"
  - *(IANA is the Internet Assigned Numbers Authority, an ICANN subdivision that manages the top of the DNS namespace, IP addresses, and protocol parameters)*

Top Level Domains

Until recently, there were 4 classes of TLDs

- **gTLD**: Generic Top Level Domains (com/net/org etc)
  - generically open to registration by anyone
- **Sponsored TLD**: .mil, .gov, .edu
  - restricted to some specific communities
- **ccTLD**: Country Code TLDs: (.us, .jp, .de, .uk, ….)
  - 2-letter codes for each country, per ISO-3166
- **Infrastructure**: .arpa (reverse DNS, e.164 etc), .int (now deprecated)

Expansion: New gTLDs

- 2012 - ICANN introduced program to establish new generic TLDs
- Delegations began appearing October 2013
- 1,154 delegated as of the end of 2022 (source: ntldstats.com)
- For additional details: newgtlds.icann.org

abogado academy accountants active actor adult agency airforce allfinanz alsace amsterdam android apartments aquarelle archi army associates attorney auction audio autos axa band bank bar barclaycard barclays bargains bayern beer berlin best bid bike bingo bio black blackfriday bloomberg blue bmw bnpparibas boats boo boutique brussels budapest build builders business buzz bzh cab cal camera camp cancerresearch canon capetown capital caravan cards care career careers cartier casa cash casino catering cbn center ceo cern channel chat cheap christmas chrome church citic city claims cleaning click clinic clothing club coach codes coffee college cologne community company computer condos construction consulting contractors cooking cool country courses credit creditcard cricket crs cruises cuisinella cymru dabur dad dance dating day dclk deals degree delivery democrat dental dentist desi design dev diamonds diet digital direct directory discount dnp docs domains doosan durban dvag eat education email emerck energy engineer engineering enterprises epson equipment esq estate eurovision eus events everbank exchange expert exposed fail fans farm fashion feedback finance financial firmdale fish fishing fit fitness flights florist flowers flsmidth fly foo football forsale foundation frl frogans fund furniture futbol gal gallery garden gbiz gdn gent ggee gift gifts gives (and many more …)

Protocol Evolution & Impediments

EDNS: Extension Mechanisms for DNS

- Radical/clean-slate designs likely infeasible (entrenched nature of DNS)
- Need framework for incremental evolution.
- EDNS - Extension Mechanisms for DNS (RFC 2671, updated in RFC 6891)
  - More than 20 years old
  - A framework for incrementally extending and evolving DNS via flags and options
  - Some defined capabilities:
    - Larger UDP packet sizes (original DNS protocol had a limit of only 512 octets)
    - Signaling and enabling new features:
      - DNSSEC, Cookies, Client Subnet, NSID, Padding, Keepalive, and more

DNS Protocol Complexity Concerns

- Is the DNS protocol too complex?
- In “The DNS Camel”, Bert Hubert, March 2018, argues that it is.
- Sheer volume of protocol specification docs
    - Hundreds of RFCs, comprising thousands of pages.
- Only deep experts can keep up (well known companies & OSS providers)
- There are lots of bad DNS implementations in the field today
    - middleboxes, proxies, set-top-boxes, printers, routers, firewalls, censorship devices, etc.
    - authors of these implementations don’t seem to have read much of the specs.
    - ossified assumptions in them often prevent the rollout of new features.
    - A large fraction of code in modern DNS resolvers is workarounds for broken servers -> DNS Flag Day efforts to improve the situation.
- Significant pushback now encountered by new enhancement proposals.

Startling example of bad deployed code

```
char resppacket[512];
unsigned int ip_address;
char *ptr=resppacket+12;

/* receive */
while(!(*ptr==0xc0 && *(ptr+1)==0x0c)) ptr++;
memcpy(&ip_address, ptr+6, 4);
```

Parsing DNS responses by pattern matching; perhaps tried to learn DNS by examining packet captures.

from Bert Hubert’s talk on “The DNS Camel”, IETF 101, March 2018

Startling example of bad deployed code

Just skip over the DNS header entirely without examining it. Then search for a compression pointer which most likely denotes the "Answer" section (because it typically starts with a back reference to the query name in the Question. Then extract the IP address from the presumed resource record at that location.

```
char resppacket[512];
unsigned int ip_address;
char *ptr=resppacket+12;

/* receive */
while(!(*ptr==0xc0 && *(ptr+1)==0x0c)) ptr++;
memcpy(&ip_address, ptr+6, 4);
```

Parsing DNS responses by pattern matching; perhaps tried to learn DNS by examining packet captures.

from Bert Hubert's talk on "The DNS Camel", IETF 101, March 2018

Non-standard Features

Non-standard features in use in the DNS today

- Many commercial DNS providers have developed proprietary features.
    - Not defined in any formal DNS protocol specification today
- Most often called “Traffic Management”
    - Global Server Load Balancing (GSLB), probe health and failover pools, M of N responses, proportional distribution, custom programmed responses, etc.
    - Zone apex traffic redirection to CDNs or other 3rd parties.

- Also known by other more colorful names
    - P. Vixie “What the DNS is not”, ACM Queue, November 2009
    - DNS more brittle, harder to debug, layer violations, cost shifting, etc.

Tussle

Long standing tussle between DNS protocol engineering purity, vs. the reality of how extensively these mechanisms are already deployed in the field.

Tussle

Long standing tussle between DNS protocol engineering purity, vs. the reality of how extensively these mechanisms are already deployed in the field.

“In the ultimate, the DNS should hold programs as well as data”

Paul Mockapetris, July 2018
“Lessons from history relevant to the future of DNS”
2018 ICANN DNS Symposium keynote

Should we standardize these?

- Some efforts are underway: Zone apex redirection now has a standardized solution (HTTPS and SVCB records - see later part of this talk)
- But other more dynamically computed responses may require more involved changes (e.g. “storing programs in the DNS”)
    - And this may encounter resistance from some commercial DNS providers who feel certain “proprietary” features constitute their secret sauce and thus competitive advantage.

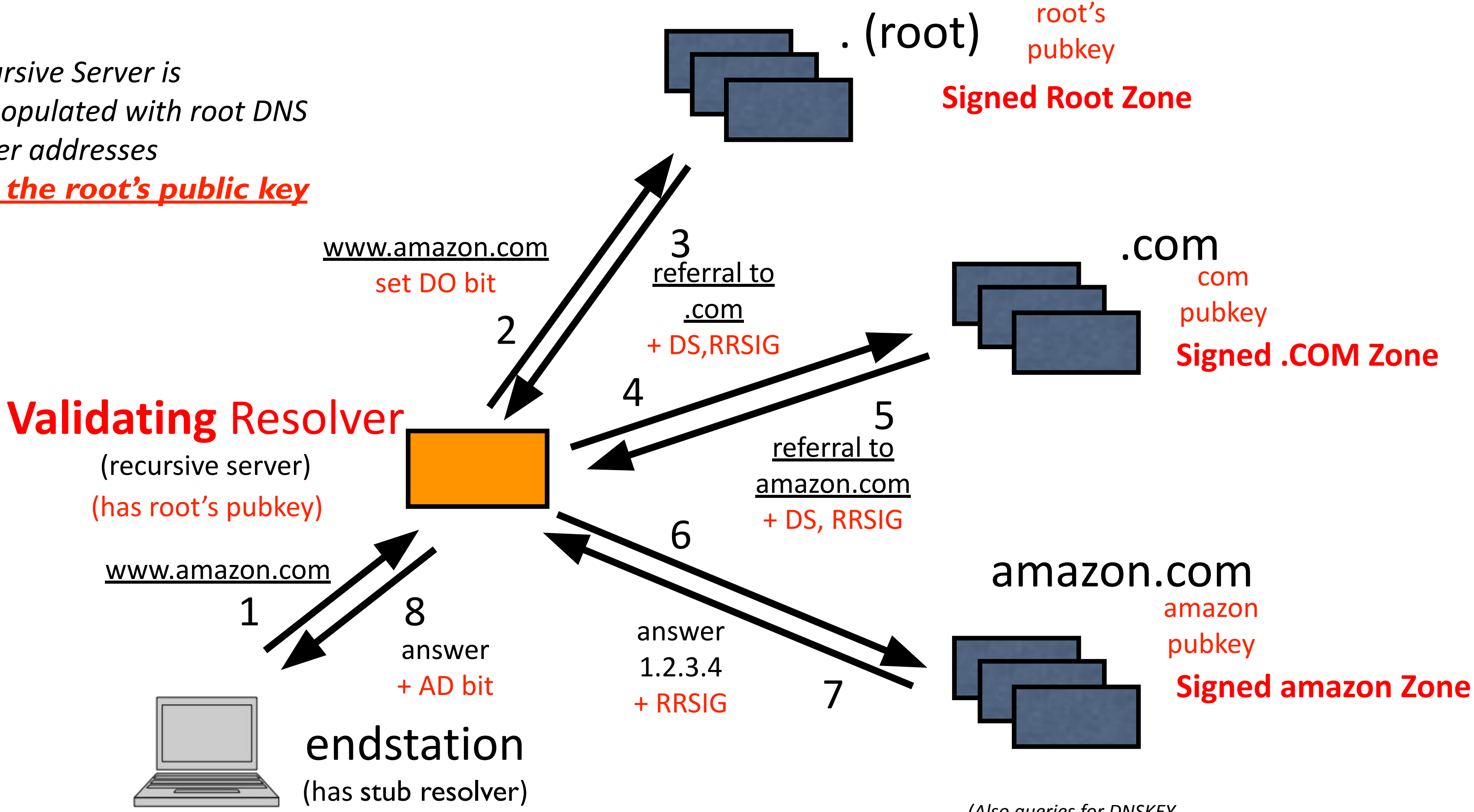
DNSSEC: DNS Security Extensions

DNSSEC at a glance

- Original DNS was not secure
  - Easily spoofed; both on-path and off-path (blind) attacks were possible

- DNSSEC: “DNS Security Extensions”
  - Core details: RFC 4033, 4034, 4035 (Mar 2005) and many subsequent specs
  - A system to verify the authenticity of DNS “data”
    - By adding public key signatures to DNS responses
  - Helps detect DNS spoofing, caching poisoning, etc.
  - Secondary benefits (realized in subsequent DANE work): securely storing cryptographic keying material in the DNS (certificates, public keys, etc) used by application protocols.

Recursive Server is prepopulated with root DNS server addresses
and the root's public key



. (root)

root's pubkey

Signed Root Zone

.com

com pubkey

Signed .COM Zone

amazon.com

amazon pubkey

Signed amazon Zone

Validating Resolver

(recursive server)

(has root's pubkey)

endstation

(has stub resolver)

1 www.amazon.com

2 www.amazon.com
set DO bit

3 referral to .com
+ DS,RRSIG

4

5 referral to amazon.com
+ DS, RRSIG

6

7 answer
1.2.3.4
+ RRSIG

8 answer
+ AD bit

(Also queries for DNSKEY and DS records are performed as needed)

```
$ dig +dnssec +multi example.com. A

;; QUESTION SECTION:
;example.com.                IN A

;; ANSWER SECTION:
example.com.    300 IN A 136.147.40.2
example.com.    300 IN A 136.147.56.1
example.com.    300 IN A 136.147.41.2
example.com.    300 IN A 136.147.57.1

example.com.    120 IN RRSIG A 13 2 120 (
                           20220709025228 20220510020206 2317 example.com.
                           IPXLmibwogovApo7ndx19Wa/WR6t74Usn9XkXwlrp0Pg
                           LbEF65MVqhv0HzwSqK/DGzVqQTEre2IE0itIRGAEmg== )
```

```
$ dig +dnssec +multi example.com. A

;; QUESTION SECTION:
;example.com.                IN A

;; ANSWER SECTION:
example.com.    300 IN A 136.147.40.2
example.com.    300 IN A 136.147.56.1
example.com.    300 IN A 136.147.41.2
example.com.    300 IN A 136.147.57.1

example.com.    120 IN RRSIG A 13 2 120 (
                           20220709025228 20220510020206 2317 example.com.
                           IPXLmibwogovApo7ndx19Wa/WR6t74Usn9XkXwlrp0Pg
                           LbEF65MVqhv0HzwSqK/DGzVqQTEre2IE0itIRGAEmg== )
```

DNSSEC Deployment Status

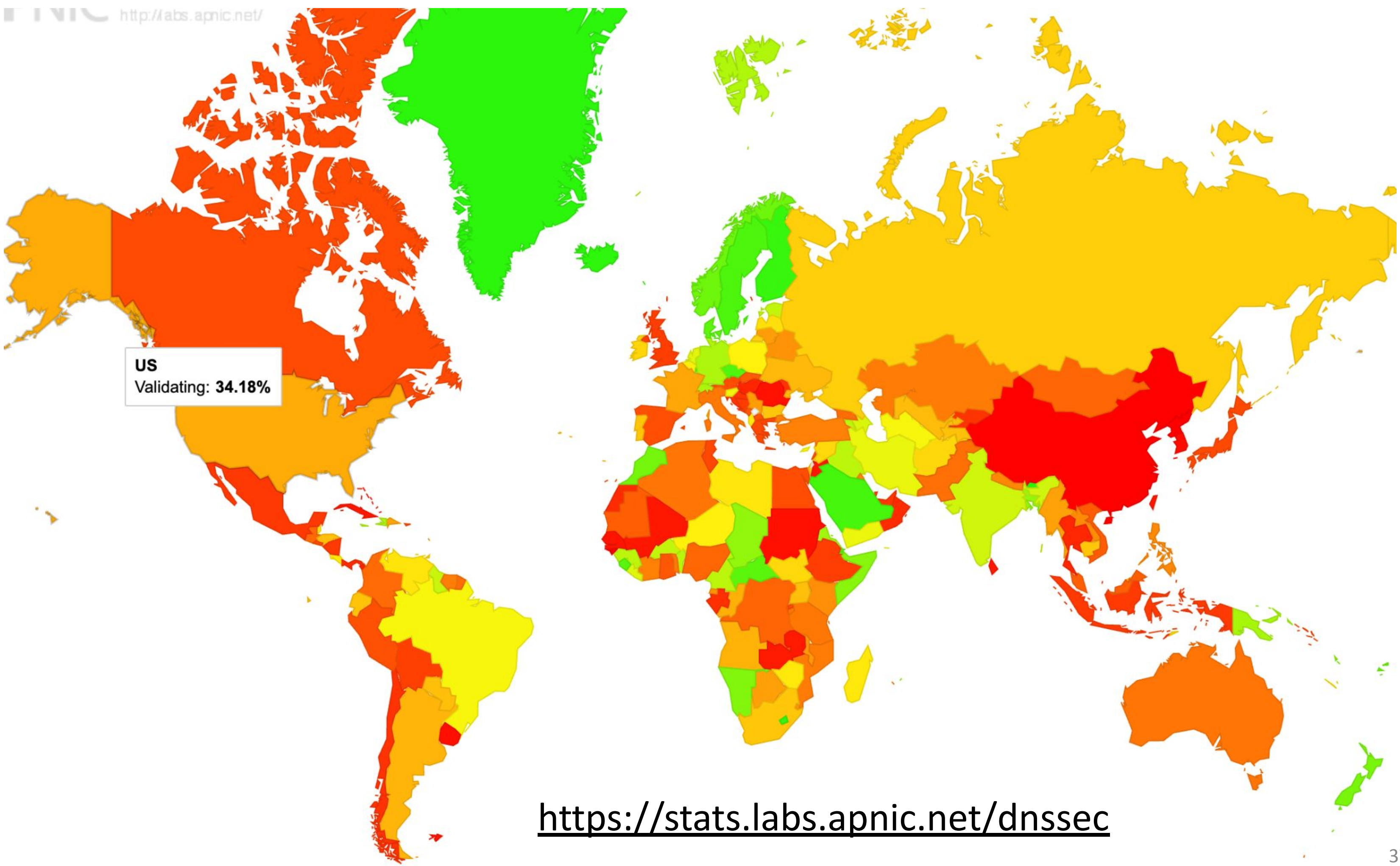
- DNS Root was signed in July 2010
- EDU, NET, COM followed in 2011/2012
- Top Level Domains overall status as of early 2020
  - All TLD: 1385 of 1515 signed (91.4%)
  - ccTLD: 176 of 304 signed (57.9%)
  - New gTLD: all are signed (contractual requirement from ICANN)
- [Details see: https://stats.research.icann.org/dns/tld_report/]

DNSSEC below the TLDs

- Levels below the TLDs are where a lot more deployment needs to happen
- Good progress in certain pockets:
  - US government agencies: ~ 85% (impetus: FISMA OMB mandate)
  - Some ccTLDs (.nl, .br, .se, etc) have very high deployment rates
- Disappointingly low adoption more generally
  - COM: 5.8 million of 158.5 million (~ 3.7%)
  - ORG and NET are similar
- Top website lists? (Alexa, Tranco, OpenDNS, ??) - also disappointingly low.

DNSSEC Validation by Resolvers

- US GOV/FISMA IT’s DNSSEC validation mandate (2014)
- Public Resolvers:
  - Google Public DNS (8.8.8.8, etc.)
  - Cloudflare Resolver (1.1.1.1)
  - Quad9 (9.9.9.9)
  - OpenDNS/Cisco, and others.
- ISPs
  - Comcast - extremely extensive deployment
- Worldwide, there is quite substantial use of DNSSEC validating resolvers.



https://stats.labs.apnic.net/dnssec

Do we really need DNSSEC?

- Common Perception: no compelling need
  - Applications are ultimately protected anyway at higher layers, with TLS and certificates
  - But defense in depth, and the need to detect attacks as early as possible, means that all layers of the stack should be cryptographically protected; that includes DNS (DNSSEC) and Routing (RPKI, SBGP, etc)

Attacks do happen - one example

- Cryptocurrency wallet company, MyEtherWallet, was victimized by a Routing and DNS spoofing attack a few years ago.
  - Hacker Hijacks DNS Server of MyEtherWallet to Steal $160,000
  - **TLS did not protect the victims**, since they just clicked through the security warnings.
  - However, the DNS spoofing ability also would have allowed the attackers to obtain real “domain validated” certificates if needed.
  - **The company soon after moved to a different cloud provider that offered both RPKI route origin validation and DNSSEC.**

DANE: DNSSEC as a PKI

DANE: potential killer app for DNSSEC?

- RFC 6698: “**D**NS-Based **A**uthentication of **N**amed **E**ntities (DANE)”
- Use of Signed DNS records (i.e. DNSSEC) to authenticate Public Keys and/or X.509 Certificates used by application security protocols like TLS, HTTPS, SMTP, IPsec, S/MIME, etc.
  - (Also see RFC 7671, 7672, 7673, 7929, 8162)
- Using a system that naturally supports namespace constraints (so that only domain owners can issue their own certificates)
- Alternative to the Public CA / WebPKI system (or can apply “constraints” on the use of Public CAs.

- New “**TLSA**” record: Stores hash of a certificate or public key for a server, that can be authenticated via DNSSEC

```
;; QUESTION SECTION:
;_443._tcp.freebsd.org.         IN      TLSA

;; ANSWER SECTION:
_443._tcp.freebsd.org.  3600    IN      TLSA    3 1 1
31EF2A4D6E285CC29A636C5171F7DA0AC69CC44CEBAF5CD039DA8CC8 1187482A

_443._tcp.freebsd.org.  3600    IN      RRSIG   TLSA 8 4 3600
20190527013359 20190512132750 17338 freebsd.org.
h6BXLidwFymOeyLyjWDfzHbsPZ5Wu7gN2LECY17Gcts4k6/rkGZdDLGu
lEOb2LXDsI3ge/NZhFsy5nXvmFDr3BZoExAH2dRotIdELT280JjrMg0J
XTJeO/izwnUER+du3k0C1r+oou81DUpfX+SFnQKOzisaXe/tKnv2NJx7
Czpz/RQ5StsjAzTBOzgkyceCNAkudXAcRTCz9YxzexJIcE0AGkXUOGEB
3e0p3Hgv6X6Y6Uy+n7H7RsKAU3R40tJ3AGi5RNvK7CMxpO2qQJS62mUP
8Sya/kk/n4gw4PtyNwRBCnM5wA0DH1DQrE/qOOA6jj8zIEC422nAvgOX pEI9kw==
```

- New “**TLSA**” record: Stores hash of a certificate of public key for a server, that can be authenticated via DNSSEC

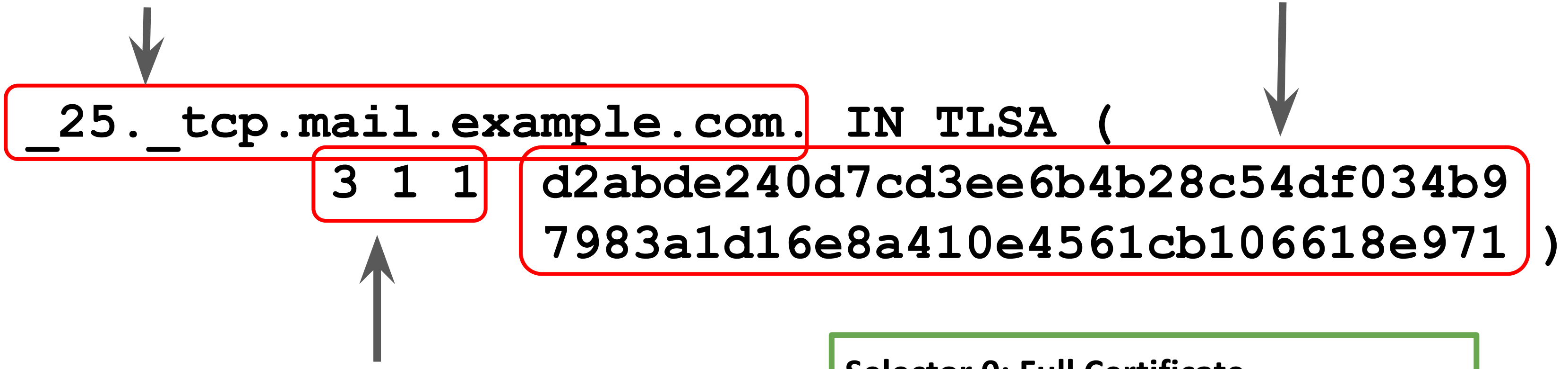
```
;; QUESTION SECTION:
;_443._tcp.freebsd.org.          IN      TLSA

;; ANSWER SECTION:
_443._tcp.freebsd.org. 3600      IN      TLSA     3 1 1
31EF2A4D6E285CC29A636C5171F7DA0AC6[illegible]C44CEBAF5CD039DA8CC8 1187482A

_443._tcp.freebsd.org. 3600      IN      RRSIG    TLSA 8 4 3600
2019052701[illegible]13359 20190512132750 17338 freebsd.org.
h6BXLidwFymOevLyjWD1zHbsPZ5Wu/gN2LECY17Gcts4k6/rkGZdDLGu
lEOb2LXDs[illegible]I3ge/NZhFsy5nXvmFPr3BZoExAH2dRotIdELT280JjrMg0J
XTJeO/izw[illegible]UER+du3k0C[illegible]+o[illegible]u81DUpfX+SFnQKOzisaXe/tKnv2NJx7
Czpz/RQ5StsjAzTPOzgryceCNAkudXAcRTCz9YxzexJIcE0AGkXUOGEB
3e0p3Hgv6X6[illegible]Uy+n7H7RsKAU3R40tJ3AGi5RNvK7CMxpO2qQJS62mUP
8Sya/kk/n4gw4PtyNwRBCnM5wA0DH1DQrE/qOOA6jj8zIEC422nAvgOX pEI9kw==
```

No need to trust Public CAs (Certification Authorities) any more?

1-Slide DANE Record Primer



port, protocol, domain name

data (hex encoded) associated with the certificate or public key

```
_25._tcp.mail.example.com. IN TLSA (
          3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                 7983a1d16e8a410e4561cb106618e971 )
```

Parameters: Usage, Selector, Matching-Type

Usage 0: PKIX-CA: CA Constraint
Usage 1: PKIX-EE: Service Cert Constraint
Usage 2: DANE-TA: Trust Anchor Assertion
Usage 3: DANE-EE: Domain Issued Certificate

Selector 0: Full Certificate
Selector 1: Public Key (could be raw)

Matching-Type 0: Full Content
Matching-Type 1: SHA-256 Hash
Matching-Type 2: SHA-512 Hash

DANE record specifies the SHA256 hash of the subject public key of the certificate that should match the End-Entity certificate. Authenticated entirely in the DNS (no PKIX involved).

Public CA issues: Unconstrained Scope

- Web PKI (or sometimes Internet PKI; misnomer)
- Apps need to trust a large number of global root CAs
- No namespace constraints! Any CA can issue certificates for any entity
- Our collective security is thus equal to the weakest.
- Furthermore, many root CAs issue subordinate CAs to their customers, again (mostly) without namespace constraints
- Excellent paper from 2013: Analysis of the HTTPS Certificate Ecosystem
    - https://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf

Public CA issues: Revocation

- Lack of effective revocation.
- Long validity periods - even LetsEncrypt is 3 months.
- CRL (Certificate Revocation Lists) - ungainly and not real time.
- OCSP (Online Certificate Status Protocol) - realtime, but privacy leaking, and not even universally used.
- Stapled OCSP (RFC 6961) - addresses OCSP privacy threat, but not widely deployed. Needs “must staple” extension too to be secure, which is difficult to deploy without wide adoption; and lastly doesn’t solve the many “non-TLS” use cases.

Public CA issues: Functional Deficiencies

- Most CAs aren’t capable of issuing anything other than the most basic capabilities.
- Public CAs today basically support only DNS names and sometimes IP addresses and email addresses as identities.
- How can we support more advanced features, such as other subject alternative name forms (URI, SRVName, etc.) to better compartmentalize the security of application services running at the same domain name?

Fundamental reliance on DNS

- The Web/Internet PKI ultimately relies on domain names. Application services are all identified by domain names. These names need to be trusted anyway.
- Domain Validated certificates are very common place.
- Even Org validated or DV certificates ultimately need a way associate an organizational identity with a domain name.
- DNSSEC provides a solution to trusting domain names. And DANE enables the secure mapping of domain names to cryptographic credentials for apps.

Public CA incidents

Too many to list comprehensively, but (WoSign/Startcom, CNNIC, Comodo, ANSSI, TurkTrust, Diginotar, Symantec ..)

- Comodo https://arstechnica.com/security/2011/03/how-the-comodo-certificate-fraud-calls-ca-trust-into-question/
- DigiNotar https://www.dutchnews.nl/news/archives/2012/11/diginotar_hack_made_possible_a.php
- https://googleonlinesecurity.blogspot.com/2011/08/update-on-attempted-man-in-middle.html
- Trustwave https://www.computerworld.com/s/article/9224082/Trustwave_admits_issuing_man_in_the_middle_digital_certificate_Mozilla_debates_punishment
- TurkTrust: https://googleonlinesecurity.blogspot.com/2013/01/enhancing-digital-certificate-security.htm
- TeliaSonera: http://www.theregister.co.uk/2013/04/16/mozilla_threatens_teliasonera/
- ANSSI: https://googleonlinesecurity.blogspot.com/2013/12/further-improving-digital-certificate.html
- Comodo: https://arstechnica.com/security/2015/03/bogus-ssl-certificate-for-windows-live-could-allow-man-in-the-middle-hacks/

Public CA incidents

(continued)

- CNNIC: https://googleonlinesecurity.blogspot.com/2015/03/maintaining-digital-certificate-security.html
- Symantec: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/Hkyg_09EDYE
- WoSign: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/k9PBmyLCi8I
- Symantec https://groups.google.com/forum/m/#!msg/mozilla.dev.security.policy/fyJ3EK2YOP8/chC7tXDgCQAJ
- Symantec: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/Hkyg_09EDYE
- WoSign: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/k9PBmyLCi8I
- Symantec https://groups.google.com/forum/m/#!msg/mozilla.dev.security.policy/fyJ3EK2YOP8/chC7tXDgCQAJ
- Digicert: https://bugzilla.mozilla.org/show_bug.cgi?id=1650910

Namespace constraints in PKI?

- Technically supported in the PKIX (Internet PKI) protocol spec (see “Name Constraints” extension in RFC 5280, Section 4.2.1.10).
- But these are very seldom used - sometimes for subordinate enterprise CAs.
    - Type specificity. Lack of criticality marking.
- Not amenable to the Internet PKI business model where every CA wants to issue certificates for a global population of customers.
- We’d need a hierarchical Internet PKI to usefully use this capability (in which case, you might as well use DNSSEC)

Certificate Transparency (CT) to the rescue?

- CT specifies cryptographically verifiable and unalterable logs of issued certificates by public CAs.
- This can be used to retroactively detect fraudulently or mistakenly issued certificates and take action.
- Band-aid. Ideally, we need to have a system that prevents these kinds of mis-issuance in the first place, and not just detect them after the fact.
- Also who operates these logs? We have yet another set of 3rd parties to trust.

CAA Records?

- CAA (RFC 8659: Certificate Authority Authorization resource record)
- Zone owner publishes a CAA record at their domain authorizing only specific CAs
- May help prevent “accidental” mis-issuance of certificates by other well behaved CAs.
- Cannot solve the malicious CA problem.
- CA issuer side check only.

How can DANE help?

- Certificates and public keys (or more typically their hashes) are stored in the DNS where they can be authenticated with DNSSEC.
- DNS has hierarchical & decentralized administration with **a single highly trusted root** (rather than many unconstrained roots).
- **Namespace constraints** are inherent.
- Much more **timely revocation mechanisms** (shorter TTLs and simple DANE record removal).
- Can use (authenticated) **raw public keys**.
- Better **suited to applications that use DNS for indirection** (MX, SRV, SVCB, ..).
- **Multi-function PKI**: can be tailored to specific app/protocol use case by defining distinct DANE record types.

In an alternate universe

- Ryan Sleevi/Google: IETF 109, recounting the early history of Internet PKI efforts.
    - *"Web PKI 0.1 was a quick hack, because “that Kaufman/Eastlake proposal” (RFC 2065/DNSSEC) wasn’t done yet, and required too much to change to be quickly usable"*
    - https://datatracker.ietf.org/meeting/109/materials/slides-109-saag-requirements-for-building-a-pki-01

DANE - what applications?

- Potentially many
- In practice today, it’s mainly used by SMTP (and to a smaller extent XMPP, and object/message security in PGP and SMIME)
- There have been efforts to make it work with the Web, but that has to date failed, because of technical disagreements about protocol details
    - “Whither DANE?” - my account of that effort, and why it failed.
    - TLS DNS Chain Extension: RFC 9102 - a DANE enabling mechanism originally envisioned for web applications, to overcome middlebox and latency concerns.

Middlebox impediments

- A general problem, but DANE specifically often involves delivering signed DNSSEC responses to client/stub resolvers
- Study: Experimental Results on DNSSEC Record Delivery (IETF 114; dnsop; July 2022)
  - https://datatracker.ietf.org/meeting/114/materials/slides-114-dnsop-measuring-dnssec-success-01
- Other potential solutions to this are: use of the TLS DNSSEC chain extension, or moving queries to a secure transport like DNS over HTTPS that is more likely to be successful traversing middleboxes.

DANE for TLS Client authentication (new)

- An emerging use case for DANE
- Authenticate client side of TLS connection with DANE
- Target use cases so far:
  - SMTP Transport Security
  - IOT Device Authentication (where today primarily private enterprise PKI is in use, but cross domain cases are hard to support today)
- See new IETF working group: DANCE (DANE Authentication for Networked Clients Everywhere)

DNS Privacy

IETF Publishes RFC 7258

- In the wake of the Edward Snowden revelations

- *“**Pervasive Monitoring** is an **attack** on the privacy of Internet users and organizations.”*
- “… that needs to be **mitigated** where possible, **via the design of protocols** that make Pervasive Monitoring significantly more expensive or infeasible.”

DNS Privacy around that time

- Original DNS protocol largely unchanged from RFC 1034/1035 (1987) is still dominant
- No encryption or privacy protections
- All DNS packets are sent in cleartext

- Prevailing expectation: **DNS data are public**, so no specific need for confidentiality.
- Note: DNSSEC does not provide confidentiality – its goal was data origin authentication (i.e. integrity of DNS data)

DNS Privacy - confidentiality important

- Realization that query privacy was also very important.
  - Edward Snowden revelations (2013) had a big impact.
- DNS queries and responses are metadata, and this metadata can reveal important information about your communications.
  - .e.g. the fact that you did a DNS lookup of a drug rehab site, may give away some clues about you and your intentions.
  - So, the DNS records for that site may be public info, but the fact that a specific user looked up those DNS entries should not be.
- Even without user metadata, traffic analysis is possible via many methods, e.g. timing & size measurements, cache snooping, etc.

DNS Privacy: data minimization (skip)

- DNS Query Name Minimization (RFC 7816)
  - Limit the amount of information about domain names visible to authoritative servers.
  - By resolvers only exposing the minimum number of labels of a domain name to those servers in the iterative DNS resolution process, and building up fuller domain names as referrals to downstream zones are followed.
- NXDOMAIN Cut (RFC 8020) & Aggressive Negative Caching (RFC 8198)
  - Increase the scope of local negative caching.
  - Minimize leakage of queries for non-existing domain names from the recursive server to other authoritative servers.

- (won’t discuss any further today – refer to RFCs for additional details)

DNS Privacy: confidentiality

- Move DNS traffic to authenticated and encrypted transports
  - DNS over TLS (DoT)
  - DNS over HTTPS (DoH)
  - DNS over QUIC (DoQ)

- Note: some non-standardized options for encrypted transport have existed for some time, like DNSCurve (D.J.Bernstein) and DNSCrypt (OpenDNS).
  - No attempts to standardize these have been made in bodies like the IETF, so they are unlikely to see wide adoption.

DNS over TLS (DoT) ~ 2016

- RFC 7858: Specification for DNS over Transport Layer Security (TLS)
- Send DNS messages over TLS (Transport Layer Security)
- New dedicated port: 853/TCP

DNS over HTTPS (DoH) ~ 2018

- RFC 8484: DNS Queries over HTTPS (DoH)
- Send DNS over HTTPS (i.e. HTTP over TLS)
- Uses the same port/transport, i.e. 443/TCP

DNS over QUIC (DoQ) ~ 2022

- RFC 9250: DNS over dedicated QUIC connections (DoQ)
- QUIC is a new transport protocol already in use by the Web
  - Runs over UDP
  - Multi-streaming with no head-of-line (HOL) blocking, etc.
- Technically, HTTP3 already runs over QUIC, so when DoH uses HTTP3 as the underlying transport, we are already using QUIC
- But there are other compelling reasons to run directly over QUIC
  - performance: remove the overhead of the HTTP layer.
  - better suited to the recursive server to authoritative server environment, where HTTP stacks are not as common.

But back to DoH …

- DNS over HTTPS arrived only recently, but is already causing dramatic changes across the ecosystem!
    - Pushed by web browsers, where a small number of companies dominate the market and can cause quick, far-reaching changes!

- TLS already provides transport encryption and server authentication. HTTPS is clearly additional overhead. So what is the actual benefit/advantage over DoT?

DNS over HTTPS (DoH) Motivations

- Two Primary motivations:
  - Allowing web apps to access DNS info via existing browser APIs
  - Preventing on-path devices from interfering with DNS operations

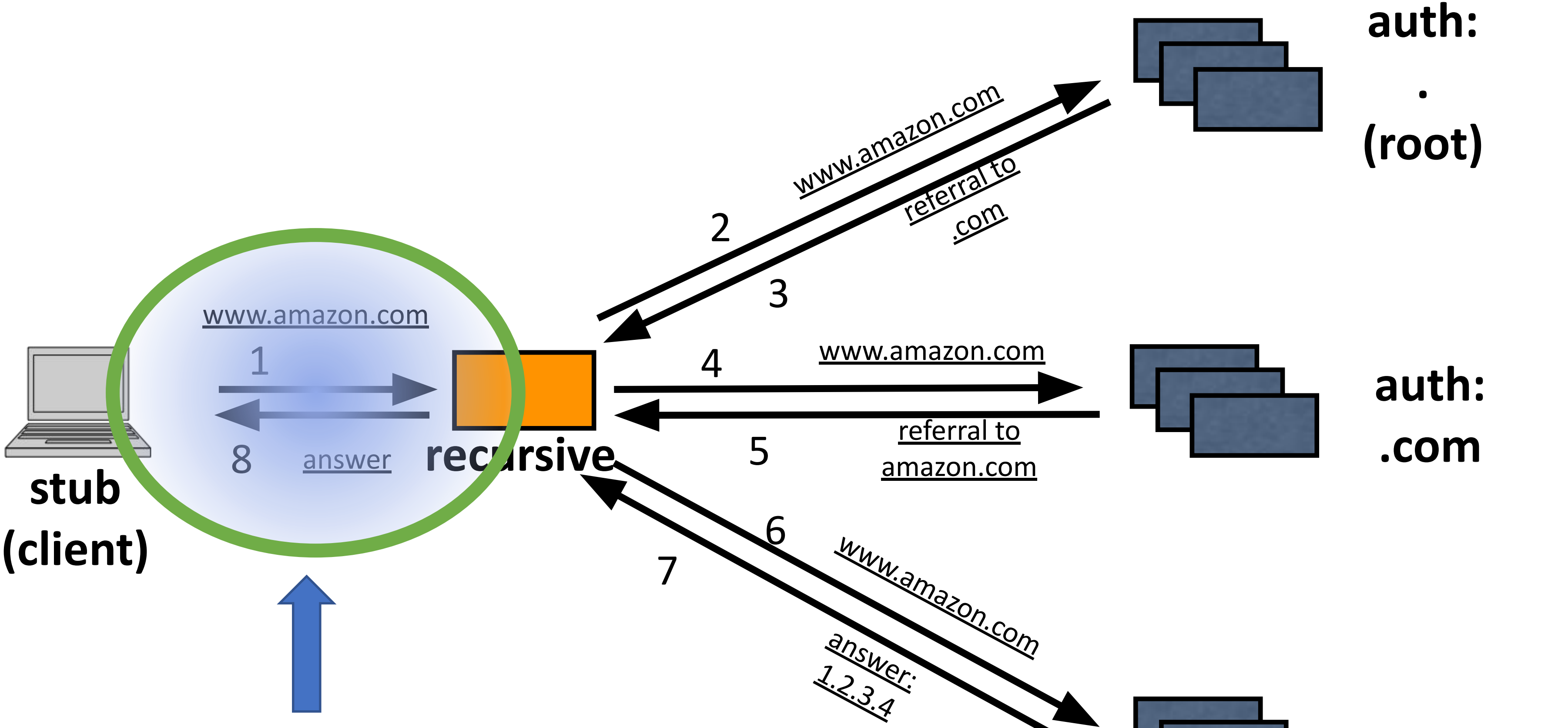
DNS over HTTPS (DoH) Motivations

- Two Primary motivations:
  - Allowing web apps to access DNS info via existing browser APIs
  - **Preventing on-path devices from interfering with DNS operations**

- Runs on the same port as Web/HTTPS, so makes it possible to **comingle Web and DNS at the same server addresses**, in a way that makes it difficult to identify, inspect, block DNS traffic, without collateral damage! (will come back to this later)

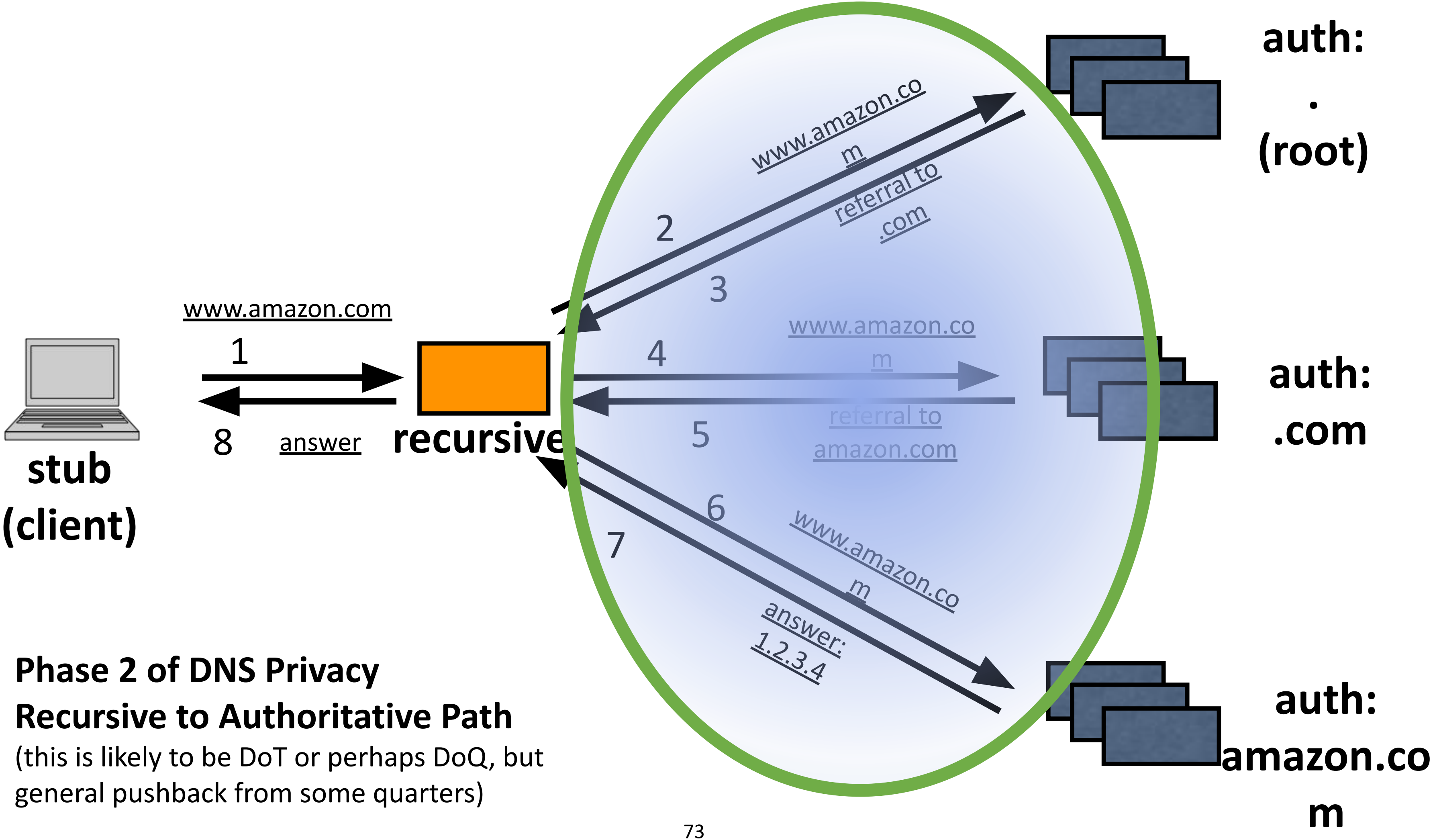
DoH vs Network Operators Tussle

- Preventing on-path devices from interfering with DNS operations
  - Actually, not just “interfering”, but also “inspecting”
- But network operators (e.g. for corporate networks, campus networks etc) often want to inspect their DNS (and other traffic) as part of their network management/monitoring/security strategy.
- DNS filtering is common for malware/abuse protection, parental controls, etc.
  - DoH may make this difficult or impossible, depending on how it is deployed.
- Other challenges: How do we support Split view/Internal DNS (commonly deployed in corporate/enterprise networks)?



Current Scope of DoT and DoH

(because this is where individual user queries can most easily be identified)



Phase 2 of DNS Privacy

Recursive to Authoritative Path

(this is likely to be DoT or perhaps DoQ, but general pushback from some quarters)

Mozilla implementation & plans

- Select Trusted Recursive Resolvers (TRR):
    - Cloudflare (1.1.1.1) initially was the sole one. There are a few more now.
- TRR Policies (some key ones)
    - Strictly limit data collection; do not sell/monetize/transfer to other parties
    - Must not filter DNS queries/responses
    - Must use Query Name Minimization
    - Must not use EDNS Client Subnet, unless resolver-auth path encrypted
- Firefox will send DNS queries using DoH to this TRR, bypassing the system's local DNS resolver completely.

Mozilla implementation & plans

- Initial plan was to default *all* Firefox users to using DoH to the Cloudflare DNS resolver.
- Backlash from various quarters.
- Now will default only US users.

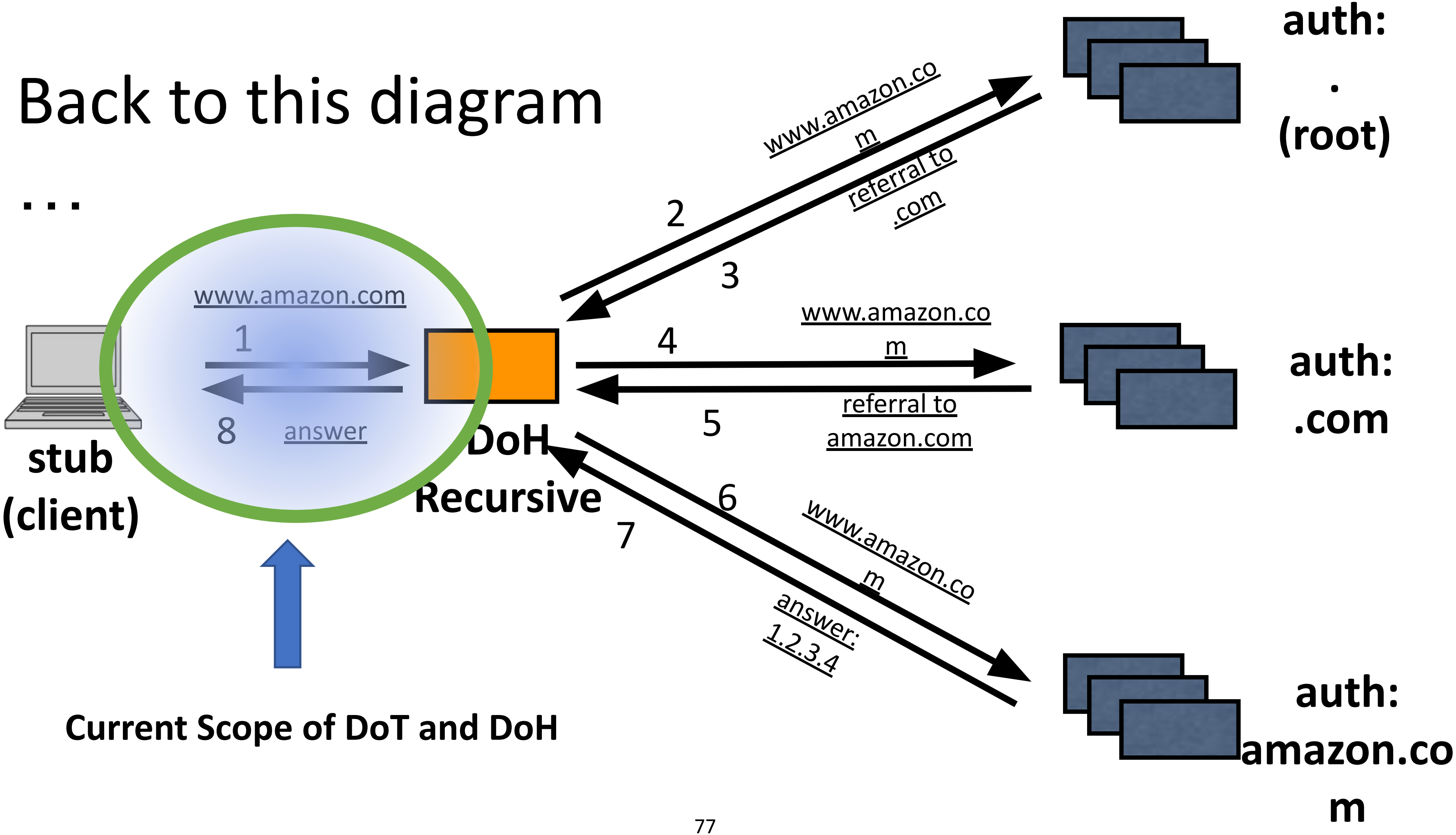
- Concession to managed network operators:
  - “use-application-dns.net” canary domain
  - Allows network operator to disable DoH for users on their network, which they almost invariably do.

Mozilla implementation shortcomings

- Doesn’t require that TRR implement DNSSEC validation.
- In fact, Firefox falls back to local resolver on DNSSEC validation failure by Cloudflare!
- Falls back to local resolver if name doesn’t resolve via DoH
  - To allow split-horizon to partially work. Doesn’t work if same name exists in both inside and outside views.
- use-application-dns.net easily spoofed by adversarial network. Users expecting DNS privacy may be in for a rude surprise, if they do not take careful steps to explicit configure their browser to avoid this.

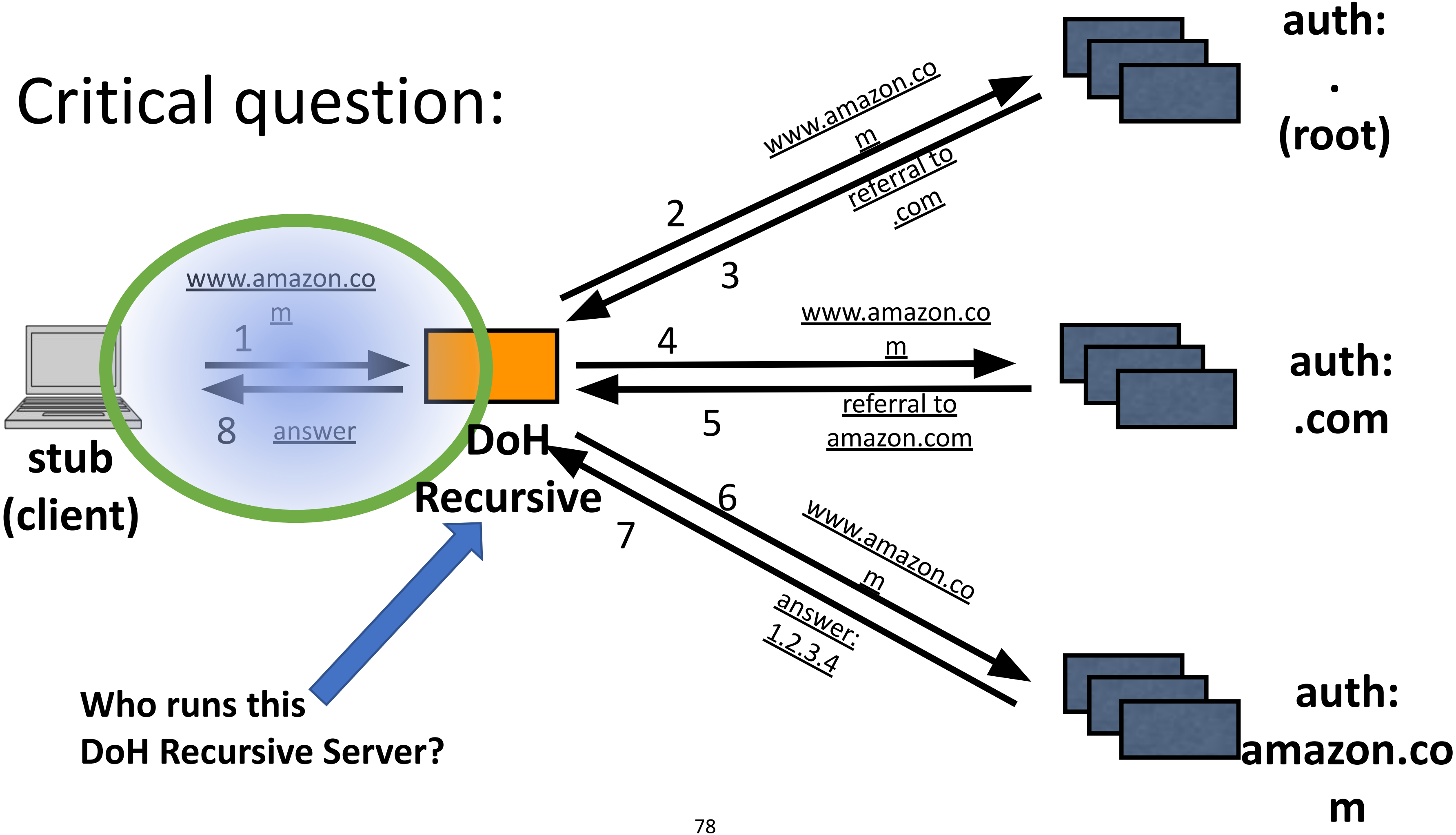
Back to this diagram

…



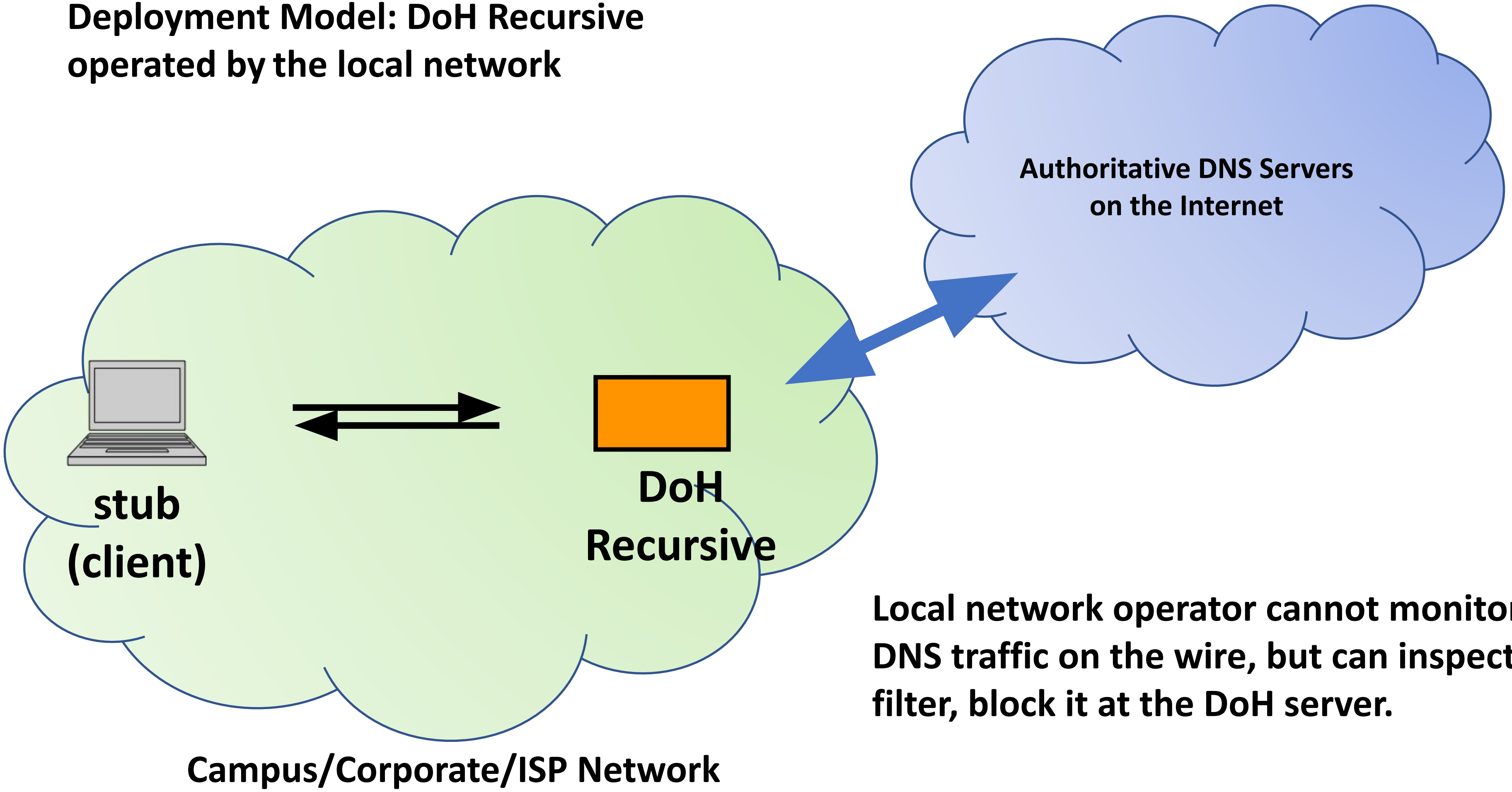
Current Scope of DoT and DoH

Critical question:



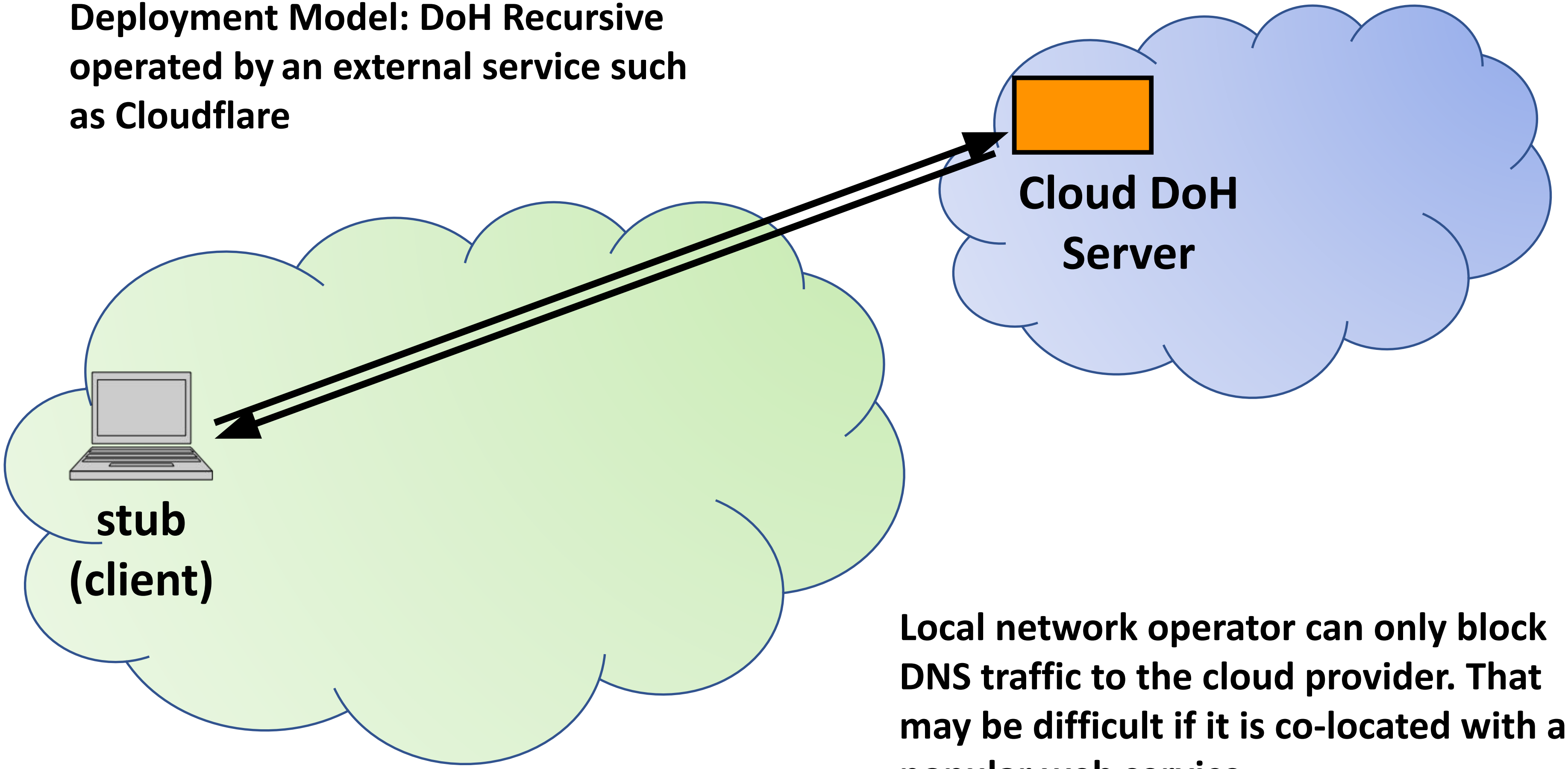
Who runs this DoH Recursive Server?

Deployment Model: DoH Recursive operated by the local network


Authoritative DNS Servers on the Internet
stub (client)
DoH Recursive
Campus/Corporate/ISP Network

Local network operator cannot monitor DNS traffic on the wire, but can inspect, filter, block it at the DoH server.

Deployment Model: DoH Recursive operated by an external service such as Cloudflare



Cloud DoH Server

stub (client)

Campus/Corporate/ISP Network

Local network operator can only block DNS traffic to the cloud provider. That may be difficult if it is co-located with a popular web service.

Is DNS a good control point for this?

- My view: not really. But managed network operators have used it in this way for a long time, and some are really upset.
- Ideally, monitoring and control is best achieved at the endpoints themselves:
    - “Moving control to the endpoints: Motivations, challenges, and the path forward” – M. Nottingham
    - https://blog.apnic.net/2019/06/11/moving-control-to-the-endpoints-motivations-challenges-and-the-path-forward/
    - We may be heading in this direction, but it won’t be easy or smooth (capital costs, re-architecture, IoT and BYOD challenges, etc.)

Not a new debate

- Network visibility of traffic
- Recall long and arduous TLS 1.3 debate about preventing on-path transparent TLS interception proxies.

Trust and Regulation?

- Many non-US folks see Mozilla's plans as distinctly US centric, where (some) ISPs are known to spy on/monetize DNS traffic.
- In Europe, ISPs are strongly regulated and aren't permitted to do this. Furthermore, there are strong privacy regulations, like GDPR that apply, not only to ISPs, but across the board.
- In their view (and actually many others), it is US internet companies that can't really be trusted.
  - (See "Surveillance Capitalism")
- Should cloud based DoH providers be regulated also?

Centralization Concerns

- If all (US) Firefox users centralize their DNS on Cloudflare, is that a good thing?

- A huge part of the strength and resilience of the Internet has come from its decentralized nature. That has been slowly changing, with the rise of CDNs, Cloud infrastructure providers, and other Big Tech services.
- See also, “Internet Consolidation: What can standards efforts do?”

Google Chrome plans?

- The largest browser by market share.
- Taking a more conciliatory approach to network operators.
- Won’t default users to using DoH, and won’t default to directing DoH traffic to their own Public Resolver
  - (Politically wise for them)
- Will try to detect DoH support in configured local resolver and then automatically upgrade to using DoH with it.
- Dec 2019: Google Chrome DoH auto-upgrade experiment:
  - https://groups.google.com/a/chromium.org/forum/#!msg/net-dev/lIm9esAFjQ0/vJ93oMbAAgAJ

Protecting dissidents & whistleblowers

- A section of the DoH advocacy camp insists that DoH is needed to offer real protection to folks like political dissidents and corporate or government whistleblowers.
- I agree that they should be protected. But we shouldn’t give them a false sense of security.
- It must be bulletproof against compromise, with no fallbacks to insecure modes. Not true for Firefox DoH.
- And most critically, we need to comprehensively plug *all* privacy leaks of domain names, not just the ones seen in DNS query and responses!

Comprehensively plugging DNS leaks

- Encrypt TLS SNI extension, which currently carries the server hostname in cleartext (“ECH” specification is coming).
- Disabling OCSP checking by client, and having servers staple OCSP responses in their encrypted handshake message.
- Hiding in the crowd, in large co-located services, or access service through a fronting server
  - Otherwise IP address alone is often enough to identify the service name.
  - But conflict with centralization concerns.
- Maybe we really need true Anonymity networks: Tor, and mix networks?
  - DoH is decidedly a partial solution, although could be one component. Also see Oblivious DoH efforts.

Academic Contributions to DNS

Academic Contributions to DNS

- DNS is a very successfully deployed protocol, and highly entrenched in the Internet
  - Thus, incremental rather than radical changes are much more likely to succeed.
  - See previous comments about barriers to protocol evolution.
  - Also see RFC 5218: What Makes a Successful Protocol?
  - (Analog to other infrastructure protocols like BGP at the inter-domain routing layer)
- Many academic (and non-academic) contributions to “Alternative” naming systems
  - Handle, Chord, GNS, CCN, NDN, Blockchain based: ENS, Namecoin etc
  - A lot of them are Decentralized vs Centralized like the DNS.
  - They have extremely niche deployments if any.

Academic Contributions to DNS

- Successes
  - Many types of Internet wide/scale measurements (DNS, DNSSEC, DANE, DoT/DoH etc)
  - Finding bugs and vulnerabilities in DNS implementations (and to a lesser extent the protocol)
  - Automated DNS zone RFC compliance: ANRP 23 prize
- Ongoing
  - Oblivious DNS (Poor man’s Tor)
  - Post Quantum Cryptography for DNSSEC signatures
    - Hash based signatures, Lattice crypto, etc [Verisign has a proposed research agenda]
- Failures
  - Protocol re-design attempts have largely failed to date.
  - One example: NSEC5 (IETF draft): make authenticated denial in DNSSEC invulnerable to zone enumeration. Technically better in almost every way, yet failed to gain traction.

Venues to engage

- Where can academic researchers engage DNS practitioners?
  - Some venues:
    - Internet Engineering Task Force (IETF)
      - Applied Network Research Workshop (ANRP)
    - Internet Research Task Force (IRTF)
    - DNS-OARC
    - ICANN
    - *NOG: Various regional Network Operator Groups (NANOG, RIPE ..), etc.
- Help us make the DNS better!

Questions/Comments?

Summary

- DNS origins & protocol overview
- DNS Industry recent changes
- DNS Protocol evolution
    - EDNS
    - Non-standard features
    - DNSSEC
    - DANE (DNSSEC as PKI)
    - DNS Privacy
    - HTTPS and SVCB
- Research areas and how academia can help

Contact information:

Shumon Huque

shuque@gmail.com

Time permitting topics

DNS Privacy vs DNSSEC

DOT/DOH/DOQ vs DNSSEC

- Mistaken assumption I’ve heard from some:
  - *“With DoT/DoH there is no longer any need for DNSSEC”*
  - This is **wrong**.

- DoT/DoH provide channel security (secure transport) of DNS messages to/from the DoT/DoH Recursive Server.
- Unless the DoT/DoH Recursive Servers performs DNSSEC validation of responses it receives, its cache can be poisoned with bogus responses, and it will happily relay those bogus responses downstream to its clients.

DOT/DOH/DOQ vs DNSSEC

- Adding DoT/DoH between the Recursive and Authoritative strengthens the picture a bit, but still does not obviate DNSSEC.
- You need to be sure that you are connecting to the right authoritative server for the zone, and it needs to be done at *all* layers of the DNS authoritative hierarchy.
- DNSSEC employs an object security model, that doesn’t require you to make sure you securely connected to the right authority servers at all levels of the hierarchy.
- If using pre-computed signatures, with the signing server offline or inaccessible to the Internet, it protects you against the compromise of authoritative servers too.
    - This is the present situation with the Root servers, and many of the high value TLDs.

HTTPS and SVCB record (time permitting)

HTTPS and SVCB record

- Problem: the DNS protocol offers no way to alias (CNAME) a zone apex to another location (e.g. a CDN or 3rd party provider).
  - This is due to the precise semantics of CNAME - it aliases a name in the DNS to another location in the tree, and all data associated with it should be looked up at the ‘target’ of the CNAME. This precludes placing a CNAME at the zone apex, but also precludes placing a CNAME at any other location in a zone if that location already has another record type (e.g. TXT, MX, etc)
- DNS has had a solution to this for a long time: the SRV (Service Location) record, but for various reasons, the Web did not adopt it.
- Hence we’ve gotten by with non-standard hacks implemented by some DNS providers (“ALIAS”, “APEXALIAS”, “ANAME”, etc.)
  - Which pose other challenges (interoperability, pre-computed signature models, etc).

HTTPS and SVCB record

- New standards based solution:
  - SVCB - a generalized, parameterized SRV like record; and
  - HTTPS, a purpose built variant specialized for the web (HTTP).
- Specification: Service Binding and Parameter Specification via the DNS (DNS SVCB and HTTPS RRs)
- Presentation at DNS-OARC34 Workshop.

SVCB record overview

```
_port._scheme.name. TTL IN SVCB SvcPriority TargetName [Parameters]
```

- Goal: bootstrap optimal connections from a single DNS query
- SvcPriority == 0: “AliasMode”
  - Enables apex aliasing (only for participating clients)
- SvcPriority != 0: “ServiceMode”
  - SvcParams: Arbitrary key-value data store:
  - TLS ALPN hints
  - Port number
  - Encrypted ClientHello configuration
  - IP hints

HTTPS

- Specialized version of SVCB for web applications (HTTP etc)
- Different RR type, but otherwise processed identically to SVCB
- Omits the _ (underscore) labels in the record owner name
- Improves compatibility with many uses of wildcarded services in HTTP