Improving DNS Privacy

and: the Battle for the Namespace

Roland van Rijswijk-Deij

NLNETLABS

Today

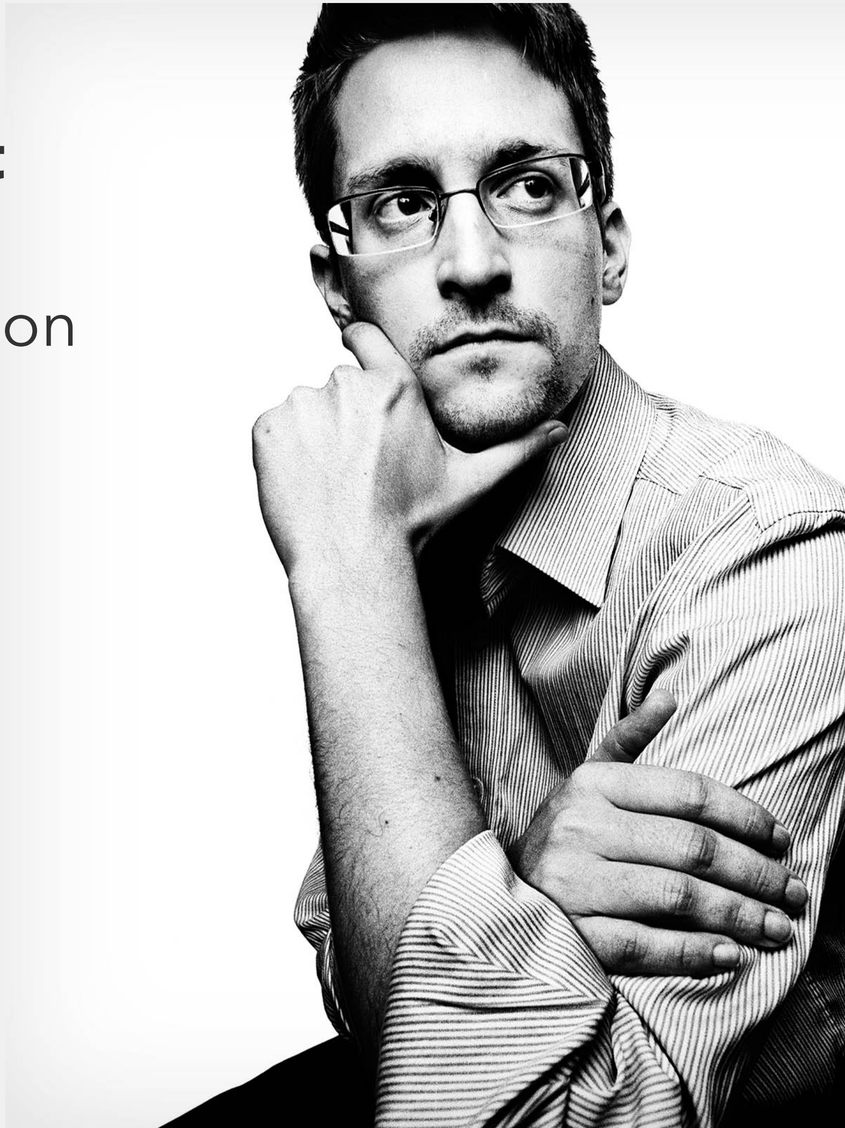
- **Who am I:**
  - **Principal Scientist at NLnet Labs** -- not for profit developing open source software for core Internet protocols and real-world research on Internet protocols
  - **Part-time assistant professor** at University of Twente (EWI-DACS)

- **Today:**
  - I will talk about privacy in the Domain Name System (DNS); my **goal** is to **show** you **how complex privacy can be in the context of real-world Internet protocols**

NLNETLABS

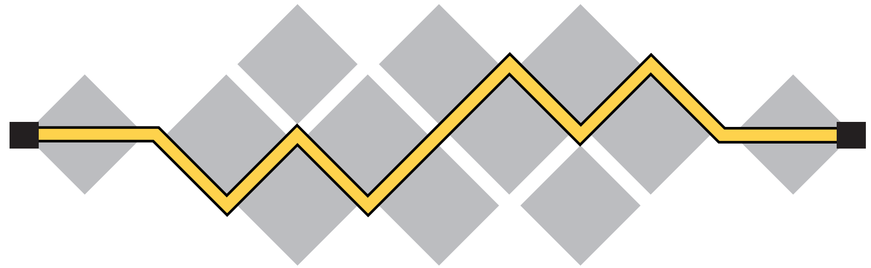
Introduction

- That the **DNS** has **privacy issues** is a **public secret**
- **Protocol from 1980s** with **clear-text** communication over **UDP and TCP**
- **Snowden revelations** just made this public secret very painful, as it turned out this was one of the Internet vulnerabilities being **exploited en masse by intelligence services** of the "Five Eyes"



IETF to the rescue!

- The **IETF took action** for many protocols **post-Snowden**
- **October 2014: establishment of** the DNS PRIVate Exchange (**DPRIVE**) working group
- **Goal**: **analyse privacy issues in** the **DNS and propose protocol changes** to alleviate these



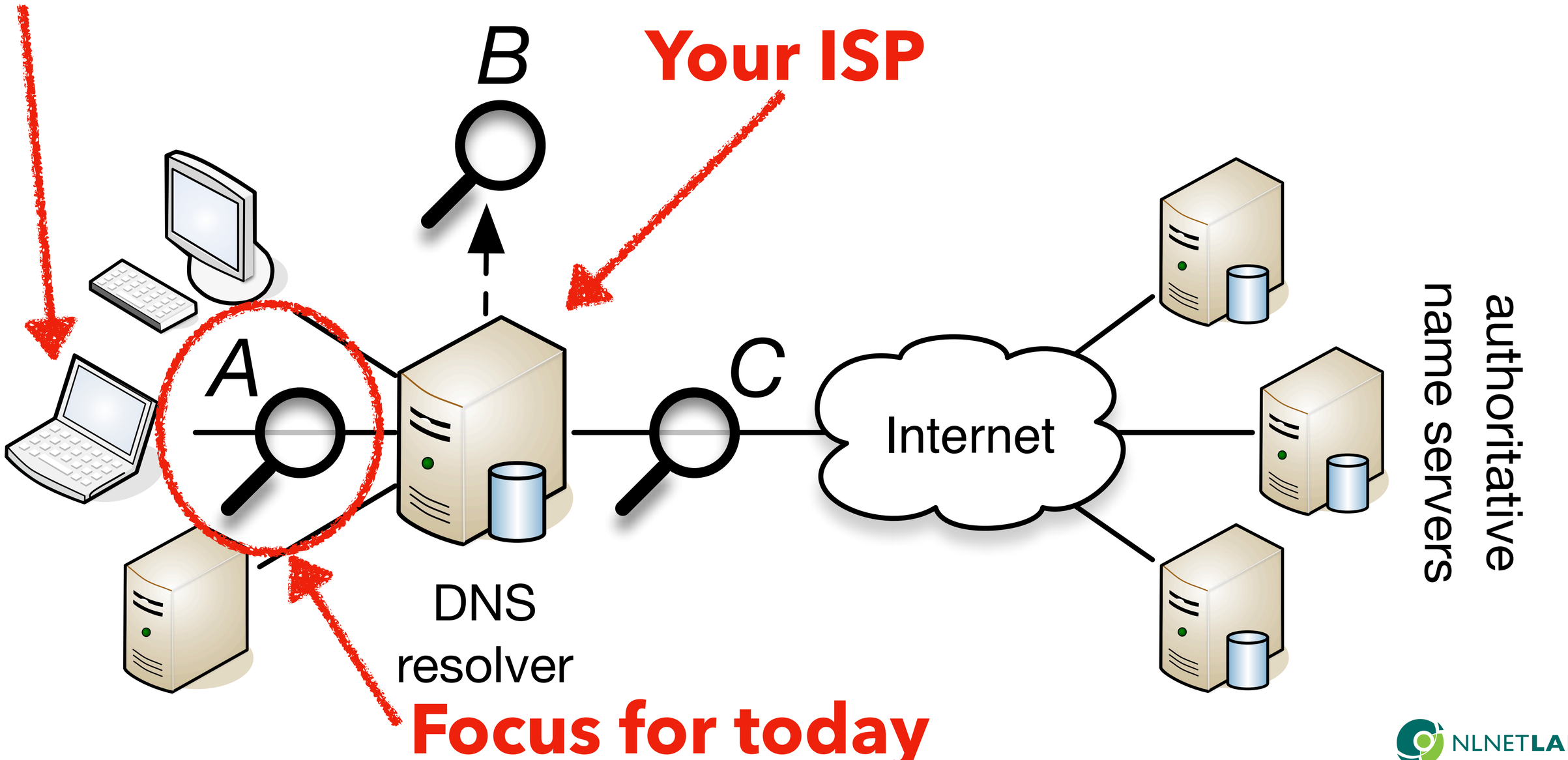
IETF®

NLNETLABS

First step: identifying problems

- **RFC 7626** gives a **comprehensive overview of privacy risks** in the whole DNS ecosystem
- Identifies all the points in the DNS ecosystem where privacy sensitive information can leak
- Today we're going to **focus on client to resolver traffic**

Recap: the DNS



Behavioural measures

- There are **two behaviour changes for** DNS **resolvers** that help privacy
- **QNAME minimisation**, where resolvers limit what parts of a query string are sent to authoritative name servers
- **Caching measures**, where resolvers can run parts of the name space locally, to limit sending, e.g., queries to the root onto the Internet
  (not going to talk about these in detail)

QNAME minimisation

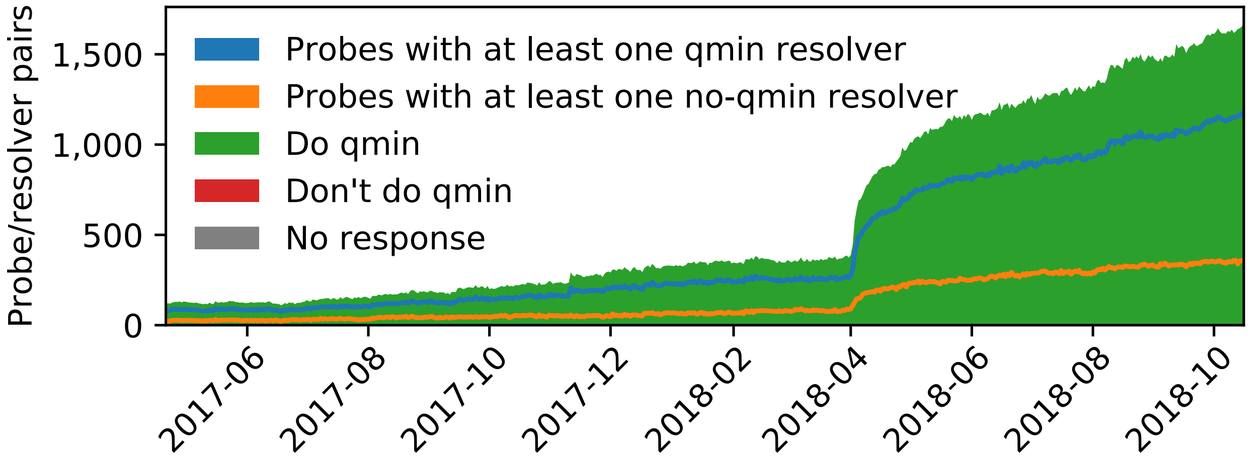
- In **"classic" DNS**, **resolver sends full query name** to every server in hierarchy → to **enhance privacy**, **only** send **necessary labels**

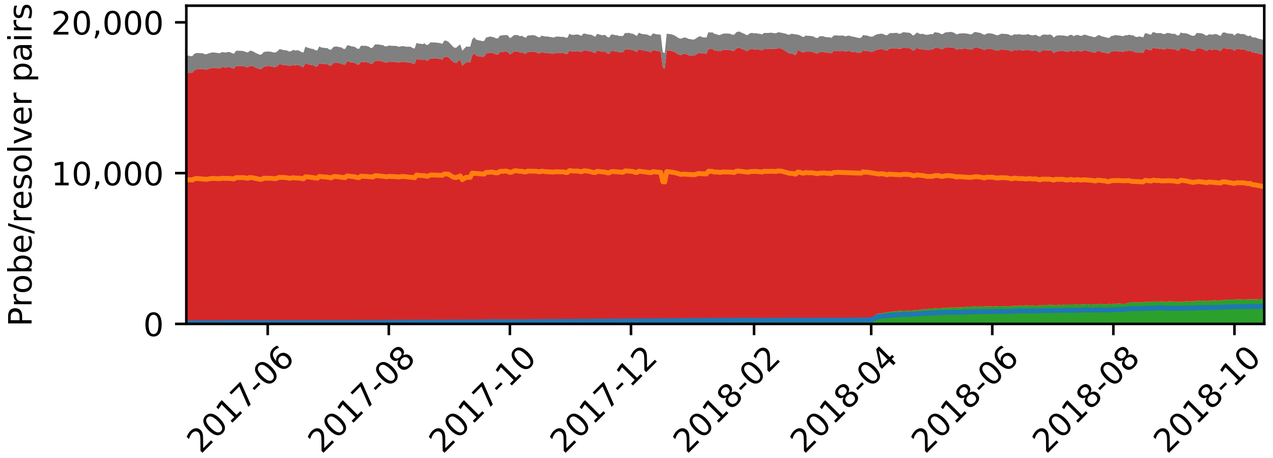
Standard DNS resolution				*qmin* Reference (RFC 7816)			
a.b.example.com.	A	→	.	com.	NS	→	.
com.	*NS*	←	.	*com.*	*NS*	←	.
a.b.example.com	A	→	com.	example.com	NS	→	com.
example.com	*NS*	←	*com.*	*example.com*	*NS*	←	*com.*
a.b.example.com	A	→	example.com.	b.example.com	NS	→	example.com.
a.b.example.com	*A*	←	*example.com.*	*b.example.com*	*NS*	←	*example.com*
				a.b.example.com	NS	→	example.com.
				a.b.example.com	*NS*	←	*example.com*
				a.b.example.com	A	→	example.com.
				a.b.example.com	*A*	←	*example.com*

NLNETLABS

QNAME minimisation

- **QNAME minimisation** is **seeing quite a bit of deployment** already
- **Supported by** e.g. **1.1.1.1 and 9.9.9.9** (among others), but also e.g. SURFnet (ISP for Dutch universities)





NLNETLABS

DNS over TLS

- **RFC 7858**: simple idea, let the **stub** talk **to** the **recursive over** a **TLS** connection

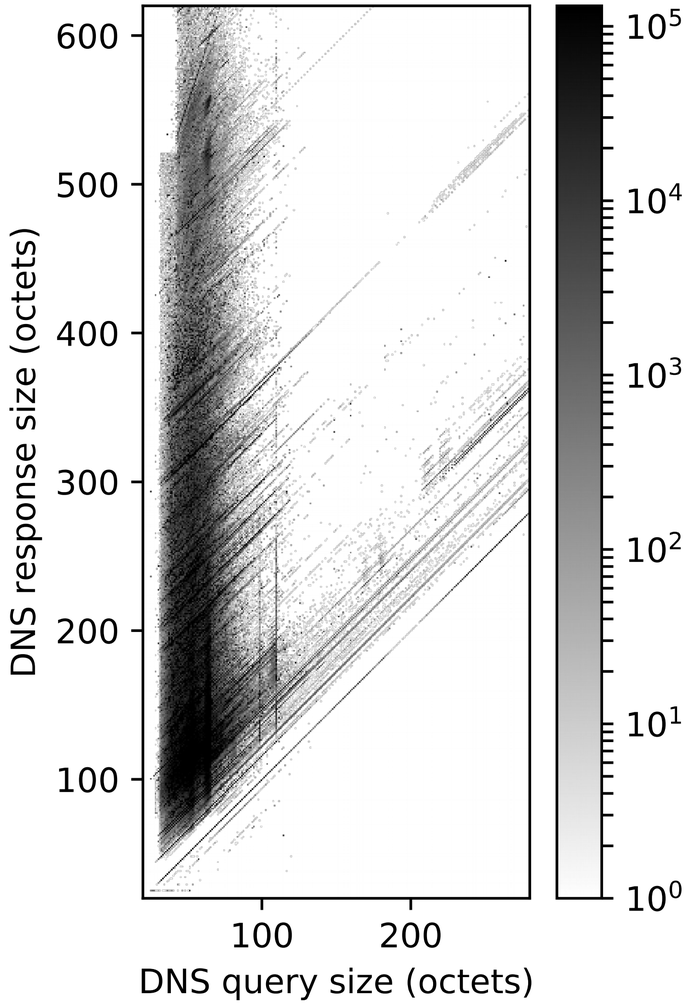
- Raises **some issues**:
  - TCP + TLS **handshake overhead**
    (partially alleviated by TCP Fast Open and TLS Session Resumption)
  - **Resource consumption** on the recursor is a potential issue
    (TCP buffers, TLS state, ...)

- **Generally** speaking, **though, works** quite **well**

Padding

- An interesting aspect of **encrypting DNS traffic** is that **padding** may be **required**
- Otherwise, the **size of queries and responses** can still be **used to profile users**!
- **EDNS0 padding** allows **stub resolvers** to **pad requests** and **recursors** that support it must **also pad responses** if the query was padded
- There are multiple approaches to padding; block-length padding seems the most sensible

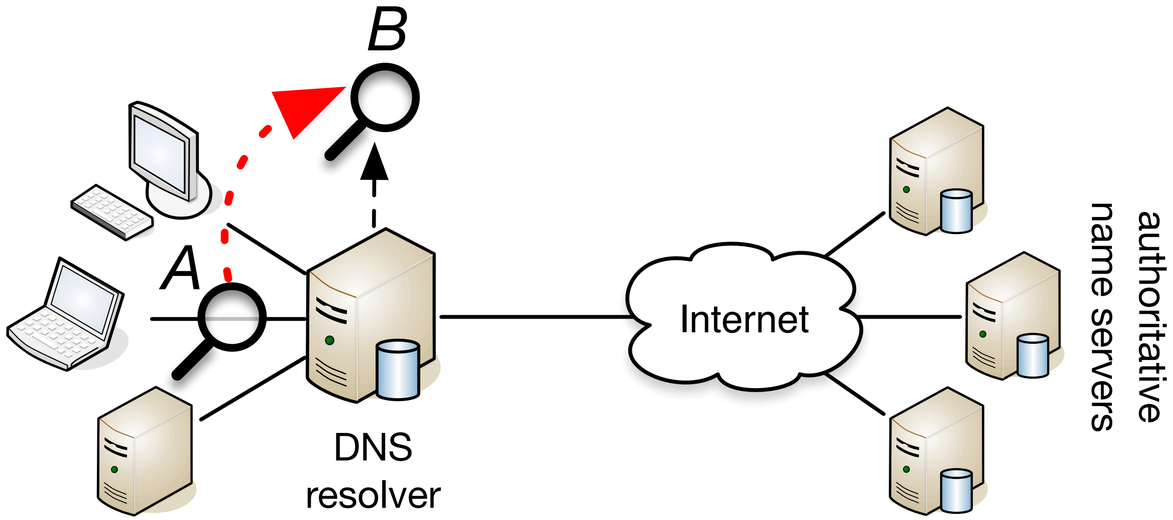
(plot courtesy of Daniel Kahn Gillmor, based on data from SURFnet)



NLNETLABS

Issues in DNS over TLS

- Encrypting DNS traffic means some on-path **security monitoring** will no longer work; **requires a shift from on-path (A) to on-resolver (B)**



- **Little experience** in production **with resource requirements** of DoT

- **Dedicated** TCP **port 853 may be blocked** on networks, making DoT unavailable

NLNETLABS

DoT implementation status

- **DNS over TLS is** already **well-supported** in recursors; **all** the **popular resolver implementations** support it
  (Unbound, BIND, Knot Resolver, PowerDNS Recursor)

- **Client support** jumped with the advent of
  Android P (DoT support, enabled by default)

- Other end users can use, e.g. **getDNS Stubby**

- **Service providers** also **widely support it** (all cloud resolvers, but also, e.g., SURFnet DNS resolvers, which use Unbound)

NLNETLABS

Next steps in DoT

- **Improve performance** by supporting, e.g., out-of-order processing
- **More support** in built-in system **stub resolvers** (slowly arriving, e.g., systemd-resolved now has support)
- Also use **TLS on recursor to authoritative path**; but how do we make this work? How to build the trust relationship (is it even possible/ necessary?)

Privacy conscious monitoring

- **Remember** that **encrypting traffic makes monitoring harder**
- Last year, **we developed a potential solution** to this:
  use of so-called **Bloom Filters**
- Tested in production at SURFnet (national research network)

NLNETLABS

Bloom Filters

- **Developed in the 1970s** to speed up database lookups
- **Highly efficient**, insertion and lookup are ~$\mathcal{O}(1)$
- **Bloom Filters are** like **a set with a probabilistic membership test**
- For a given Bloom Filter B and an element n, we can test the following:

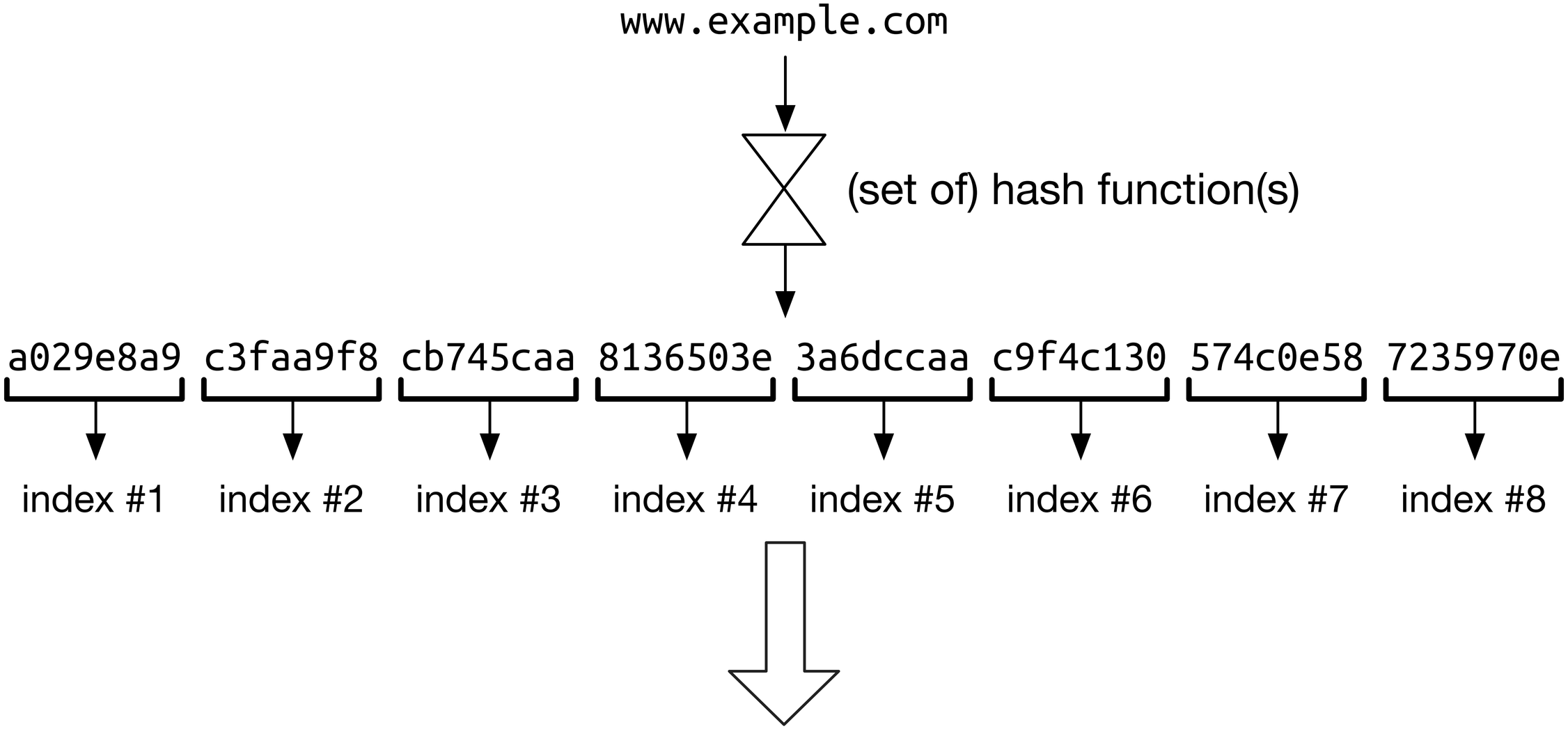
$n \in B?$

no → n is **guaranteed not** to be **in** B

yes → n is **highly likely in** B, with a **small probability** p_ε of this being **a false positive**

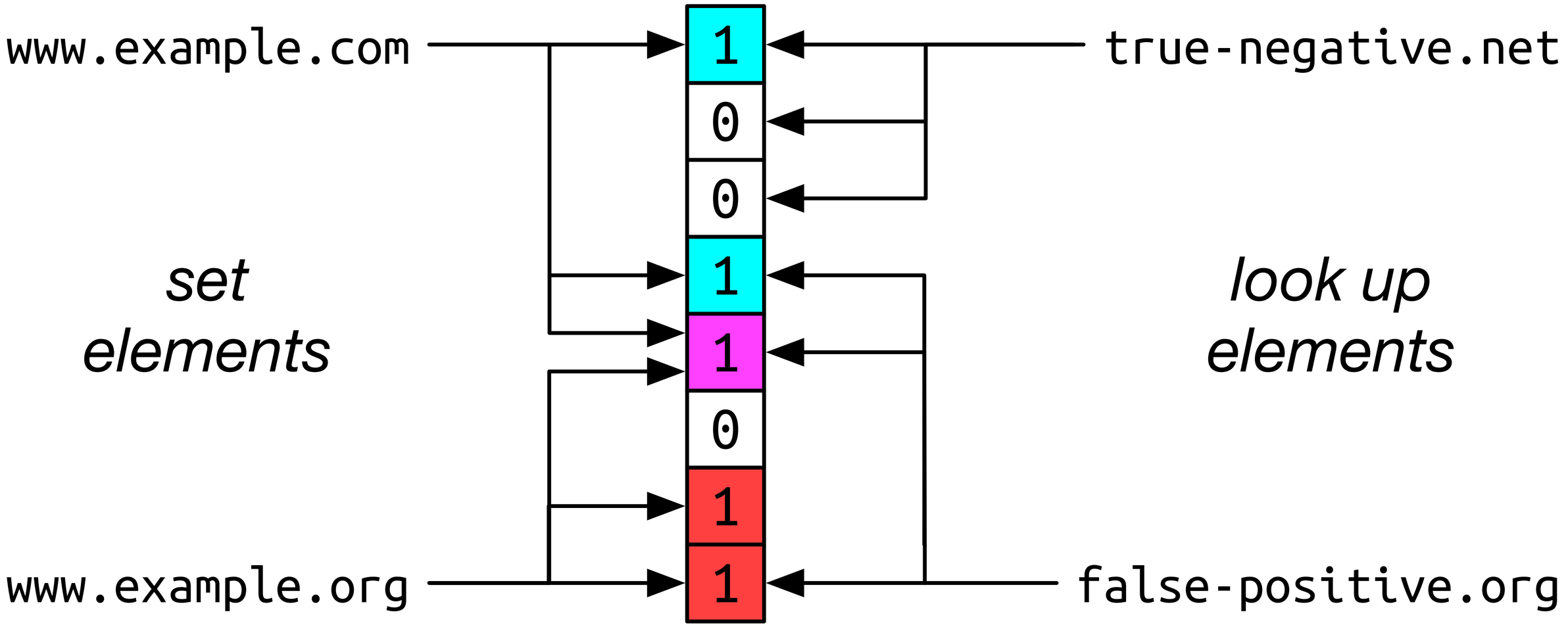
NLNETLABS

Bloom Filters



NLNETLABS

Bloom Filters



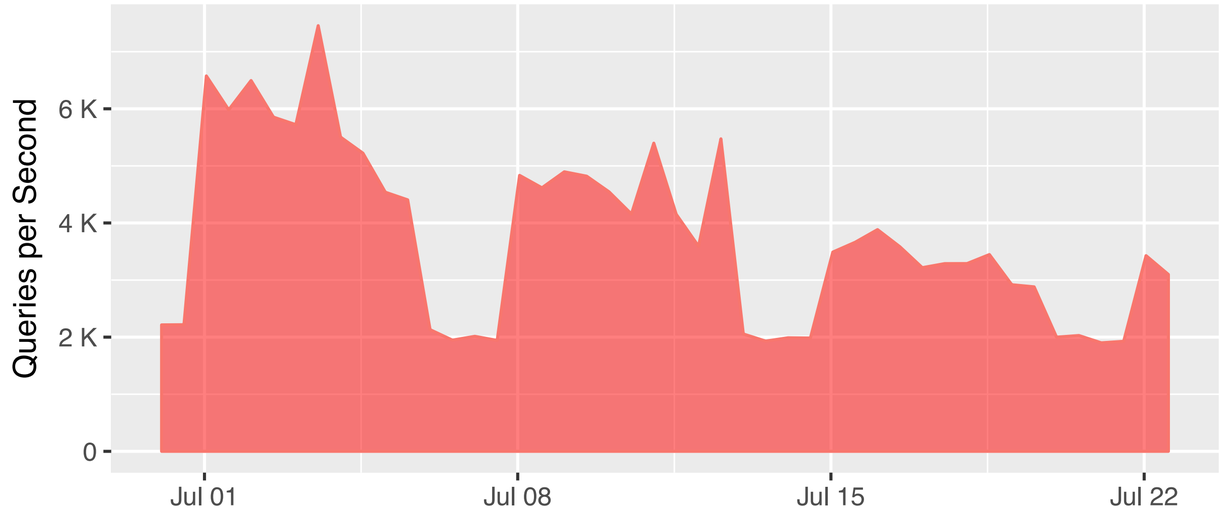
NLNETLABS

The idea

- Insert all queries from clients of a resolver into a Bloom Filter
- Then, we can check *if* a name was queried for, but *not by whom* and also *not exactly when*; this is sufficient for network-level threat monitoring
- **Privacy properties** of Bloom Filters:
  - **Non-enumerable**
  - By **mixing** queries from many **users** in a single filter, tracking becomes harder
  - Due to **mathematical properties** of Bloom Filters, we can **combine different filters**, so we can **further aggregate data over time** (making it even harder to track user)

Real-world tests

- We **tested** this **for three weeks** on busy DNS resolvers at SURFnet
- We studied **three use cases**:
  - Detection of so-called "**Booters**"
  - Hits on **e-mail blacklists**
  - Hits of high-value indicators-of-compromise for the so-called **National Detection Network**

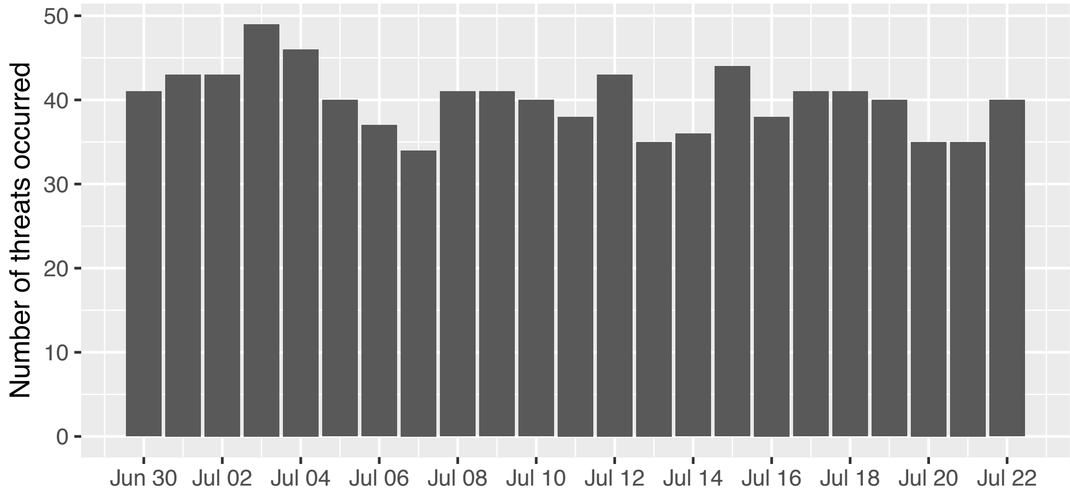


NLNETLABS

National Detection Network

- NDN is **managed by** the Dutch National Cyber Security Centre (**NCSC**) and is supposed to have **"high value" indicators-of-compromise** (from **e.g. intelligence services**)

- **SURFnet could** previously **not monitor for threats** reported in NDN because monitoring **DNS traffic** was considered **too privacy sensitive**



- **With Bloom Filter approach** it was **now possible**, and we **found actual compromises** (e.g. WannaCry infected machine)

NLNETLABS

Future of Bloom Filter solution

- **First version** of code already **released as open source** https://github.com/SURFnet/honas
- **SURFnet** is **planning to take** this **into production**
- **Future integration in** NLnet Labs **open source software** to make this approach more widely available and **easy to deploy**
- **Proof that security and privacy *can* go hand in hand!**

NLNETLABS


D'OH...
NUTS!
DONUTS
Mmm...

DNS over HTTPS

- **Google had experimental "DNS over HTTPS" for ages**; using their own REST protocol, **seemed abandoned** (nobody used it)
- Then an **IETF draft** was published, and **things started moving... FAST!**
- **DoH working group** formed in **September 2017**, **draft** adopted **October 2017**, **RFC 8484** officially **published October 2018**
- **Incredibly fast for the IETF**; lot of momentum behind this idea

DoH basic outline

- DoH simply sends **Base64-encoded wire format DNS datagrams** over either **HTTP GET or HTTP PUSH**
- **Two modes** of operation:
  - **Dedicated:** the service end point **only** functions as a **DoH DNS resolver**
  - **Mixed: DNS** traffic is **mixed into other HTTP traffic**
- DoH **server configured as** a **URI** end point in the client

NLNETLABS

DoH, where did it come from?

- **Browser community want**ed a **web-style API to** access **DNS**
- **Argumentation** browser community uses to push for it:
  - **Enhance privacy** of browser users (encrypted transport, mixing with HTTP traffic), arguing that **adoption of e.g. DoT is too slow**
  - **Port 443 does not get blocked**, so can circumvent traffic filtering
  - **Improve user experience** by reducing latency **(really?!)**
  - **Longer term: new features** (JSON, Server Push, **"resolverless"**)

Issues with DoH

- The **rest of** this **talk will focus on issues with DoH** in several dimensions

- **Why? Because DoH may have far-reaching consequences for the DNS and the Internet**

- Dimensions we will look at:
  - Issues with privacy
  - Issues for network operators
  - Impact on the DNS name space

NLNETLABS

DoH and privacy

- **Proponents push** DoH **arguing privacy**; there are **issues with** that **claim**

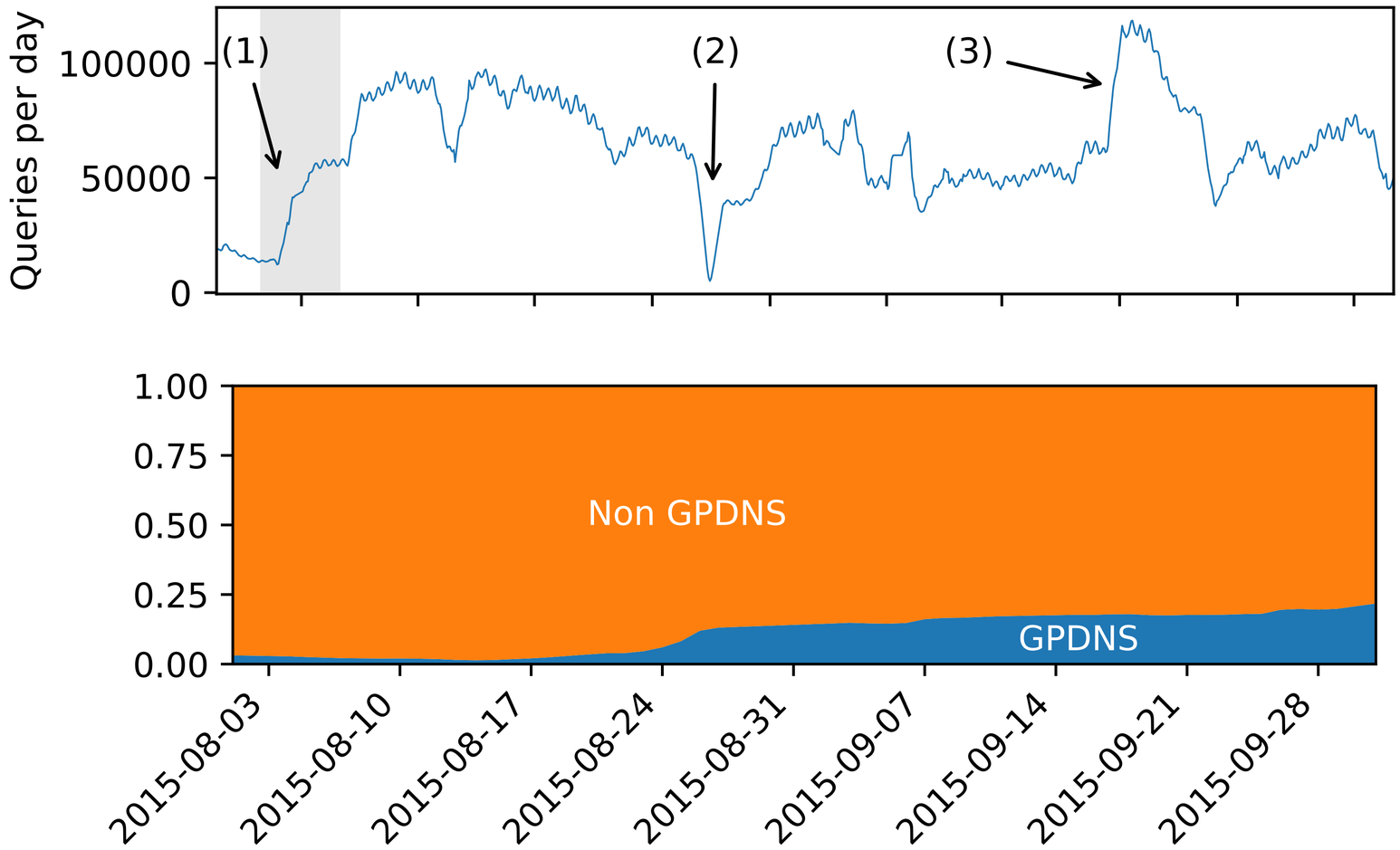
- **DoH imports** all of the **privacy issues of** the **HTTP** ecosystem into the DNS resolution process (**e.g. user agent profiling**), which has sparked a new Internet draft to address this

- DoH **proponents** appear to **advocate** that a **"public trusted recursive resolver" (TRR) is always better**. This is **simply not true** in many cases, **consider e.g. EU citizens** who are **protected by** the **GDPR** in relation to their ISP.

DoH and privacy

- **Browsers** appear on the cusp of **forcing DoH** on users
- Mozilla has **DoH** support **in Firefox since version 61**, **still disabled**, but... **considering** to **enable** it **by default**, and their **default TRR is** currently **CloudFlare**
- **Other browsers will** surely **follow** (I'm betting it's only a matter of time before Chrome will start using DoH towards 8.8.8.8 by default)
- **Users** are **highly unlikely to turn** this **off if** it's the **default**, experience with users switching to 8.8.8.8 illustrates user inertia on this

Side step: user inertia viz. DNS



Graphs show Google Public DNS use in Ziggo's AS after a DoS attack on their resolvers

Takeaway: once users change their config, they never go back

(graph from [1])

[1] W.B. de Vries, R. van Rijswijk-Deij, P.T. de Boer, A. Pras. Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google. In Proceedings of the 2018 Network Traffic Measurement and Analysis Conference (TMA 2018), Vienna, Austria, 26-29 June 2018.

NLNETLABS

DoH and performance

- Remember DoH proponents cite **"performance" as reason to deploy?**
- **Firefox** put **"classic DNS" and DoH side-by-side** (blog here)
- Here are the **weasel words from the blog**:
  *"The **slowest 20%** of DNS exchanges are **radically improved** [...], while the **majority** of exchanges **exhibit a small tolerable amount of overhead** when using a cloud service. **This is a good result**."*
- A "small tolerable amount of **overhead**" is an **average of 6ms per query!**

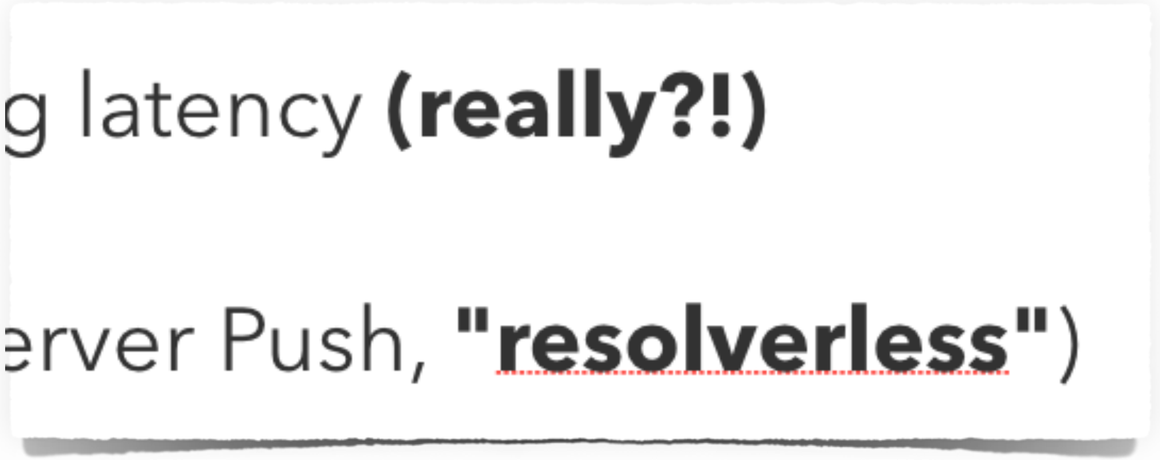
NLNETLABS

DoH and network operators

- Where **DNS over TLS may require** operators to **re-think security monitoring, DoH makes it impossible**
- Use of **DoH circumvents any local security policy** for the DNS
- Use of **DoH is (almost) impossible to track**, especially in mixed mode
- **Security officers** can look forward to **having to wrangle browser configs for managed desktops** to disable DoH and stop users from turning it back on

DoH and the DNS name space

- The **biggest** expected **impact may not be** the most **obvious**
- **Remember** that word **"resolverless"** a few slides back?
- Deployment of **DoH may radically change the DNS name space** as we know it
- **Why?**



NLNETLABS

DoH and the name space

- **Browsers vendors** and others have **floated** the idea of a **"repository of TRRs" for** looking up **specific parts of the name space**
- **Imagine a cabal** very much **like** the **CAB Forum** for the X.509 Web PKI **deciding on** a common **TRRs** in browsers (and in the future OSes too)
- Suddenly, they **decide how names are resolved**
- **Who ever gave** these folks **the right to** make **this decision?**
  What about the **multi-stakeholder** model for **Internet governance?**

NLNETLABS

DoH and the name space

- **Imagine** what this might mean!
- **Parts of the name space** are directly **resolved through** browser-**embedded TRRs**, **circumventing** the current **DNS** hierarchy
- **Next step: ICANN** and the current DNS hierarchy become **obsolete**
- What about the **"level playing field"?** How do I claim my name?
- Facilitates **further centralisation of the Internet**, and even **stronger monopolies for** certain **big players**

DoH and the name space

- **Current DNS operators** are **heavily invested in** an infrastructure that does **UDP** really well, **and** also handles **a bit of TCP**
- For **resolver operators**, it is relatively **simple to** also **support DoT**
- **DoH is a game changer**, it has a relatively **low bar of entry for players** that are already heavily **invested in** the **HTTP** ecosystem, but requires **major re-engineering for "traditional" DNS players**

What will the future look like?

- **No reason to attribute malice** to the browser folks, they are **probably** just **trying to do** what they think is **"the right thing for privacy"**
- That **"right thing"** may have **unintended and irreversible side effects**
- Because it is **tilting thinking about** how we view **the name space**
- This has **not happened** in earnest **for over 30 years**
- So we should be **pay**ing **close attention!**

Thank you! Questions?

nl.linkedin.com/in/rolandvanrijswijk

@reseauxsansfil

roland@nlnetlabs.nl

NLNETLABS