# DNS over HTTPS

## IETF 105 モントリオール

@KenjiBaheux / Sept 2019
Product Manager  @ Chrome - Web Platform, Google

# DoH @ IETF 105

**A**pplications **D**oing **D**NS **BOF:**

- Mozilla’s perspective
- Chrome's perspective
- Non-browser apps doing DNS
- DoH Preference Hints for HTTP
- DoH BCP
- DoH Push

**DNSOP:**

- DNS Resolver Information

# Role of the Domain Name System (DNS)

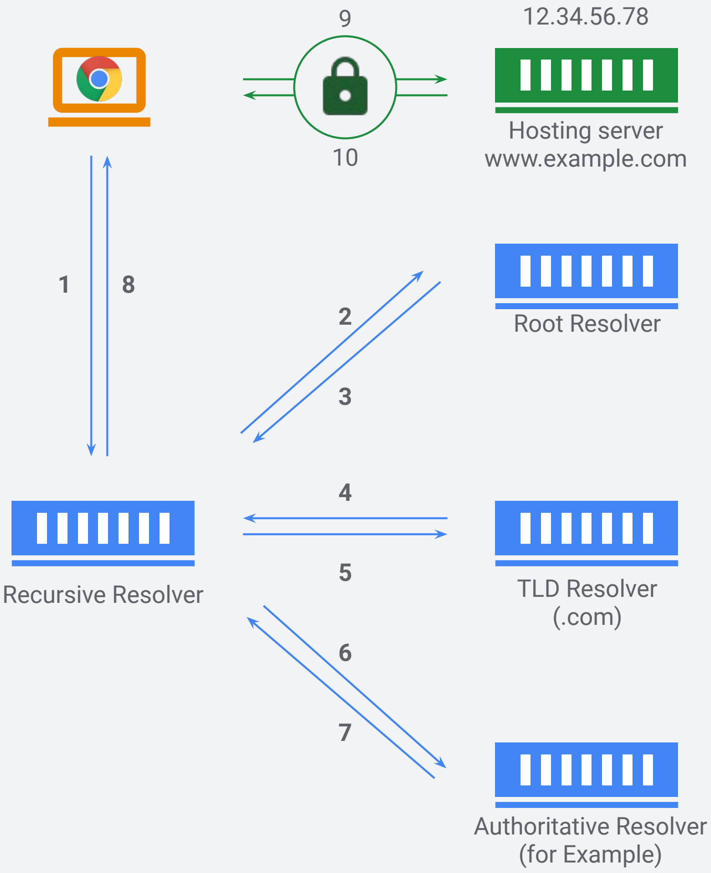
# Domain Name System

DNS is what allows your **browser** to find which **hosting server** it needs to contact when you want to visit a website, e.g. www.example.com

The browser asks[1] the **recursive resolver** (RR) to resolve the IP address for www.example.com. RR is usually provided by the ISP or network operator.

RR asks[2] a **root resolver** to find[3] the **TLD resolver** responsible for .com addresses. RR asks[4] the TLD resolver to find[5] the **authoritative resolver** for www.example. Finally, RR asks[6] the authoritative resolver to find[7] the IP address of the hosting server. The IP address is sent[8] back to the browser which can now talk[9-10] to the hosting server for www.example.com

Note: this mapping is temporarily cached by the browser to speed up future connections.



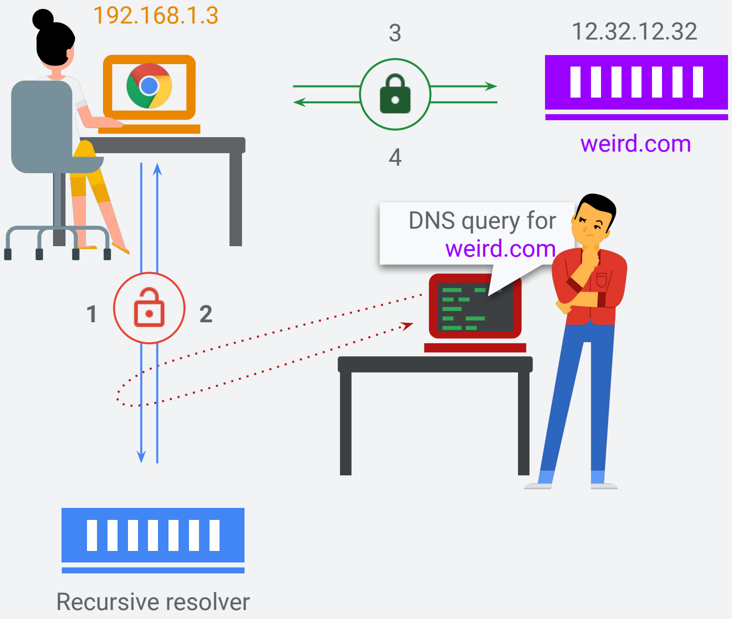
# Problems with DNS

# Lack of Confidentiality

# Privacy leak

The **connection** to the recursive resolver is **not private**.

A **bad actor** can find out which websites other users are browsing by **observing the DNS queries**, i.e. the computer at 192.168.1.3 is asking for the IP address of weird.com



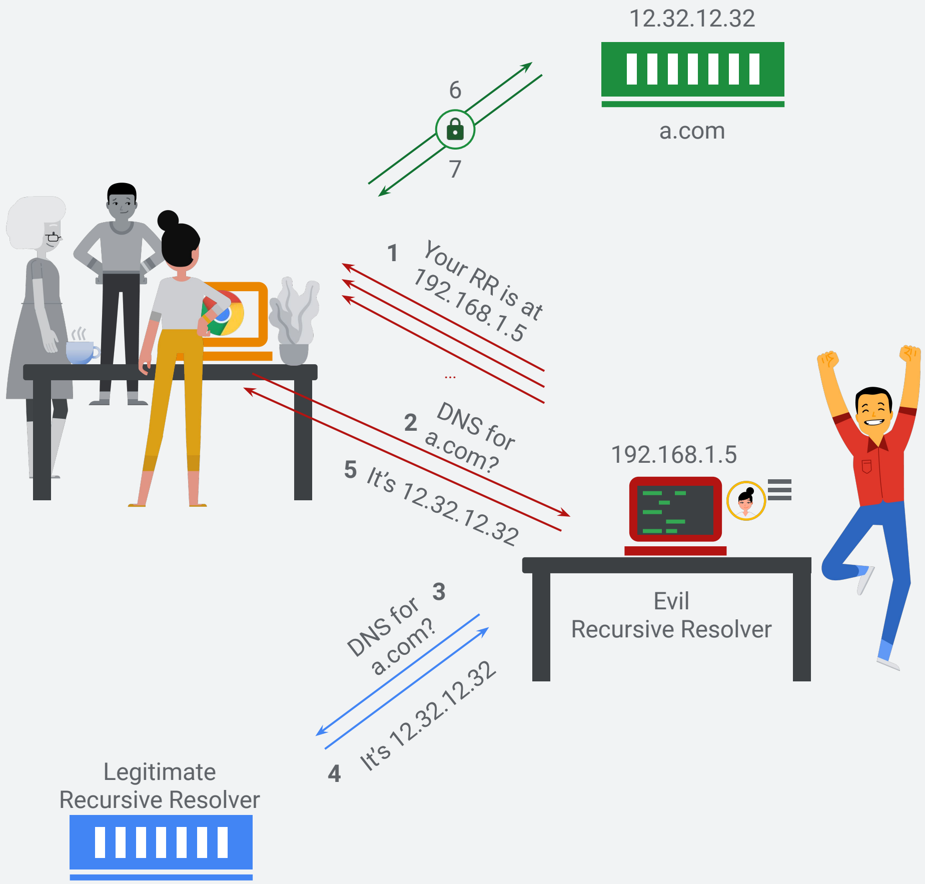
# Lack of Authentication

# Talking to strangers

The **connection** to the recursive resolver is **not authenticated**. This means that your browser has no way to confirm the identity of the recursive resolver.

A **bad actor** connected to a coffee shop's public WiFi can force (step 1) other users to use a recursive resolver that they control.

This could be their own laptop, running a simplistic recursive resolver set to capture logs and build **profiles of users**.



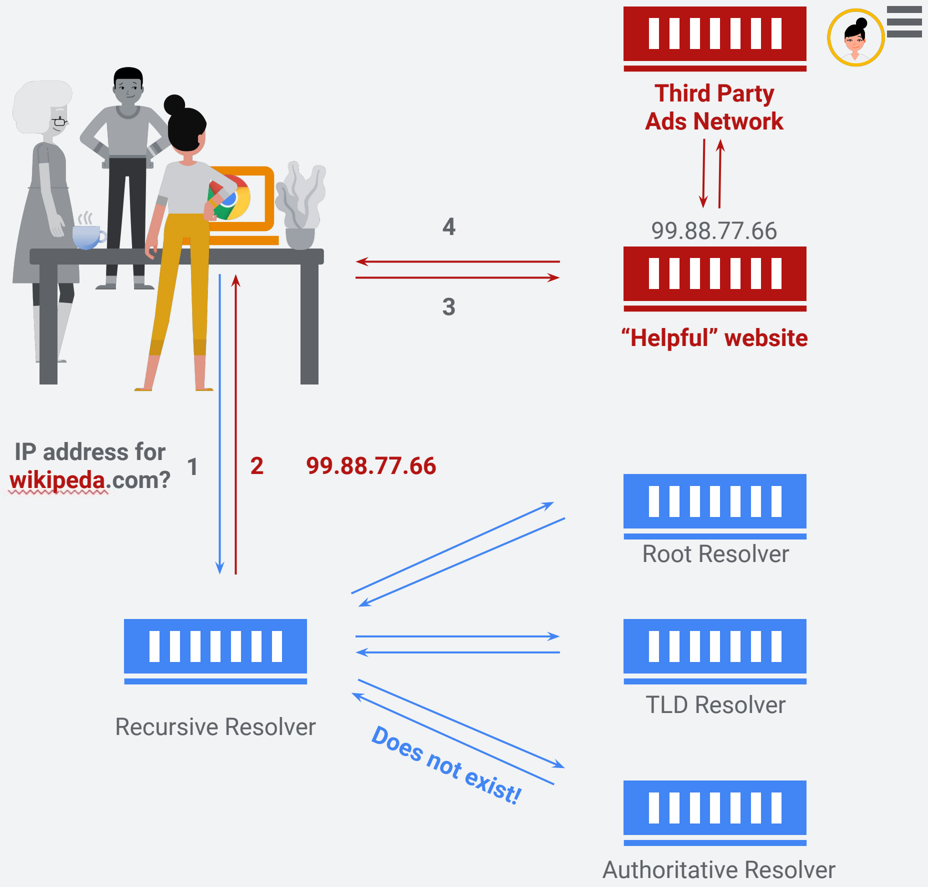
# Lack of Integrity

# Truth or lies?

The DNS responses are **unsubstantiated claims**. Your browser has no way to know if the IP address it got back is the truth or a lie.

In other words, a recursive resolver can send back whatever it wants.

For instance, if **the user** mistypes wikipedia as **wikipeda**, the recursive resolver can ignore the “**does not exist**” response it got from the authoritative resolver and send the user to a **“helpful” website ridden with ads**, allowing an **unscrupulous third party** to build a **user profile**.



# DNS-over-HTTPS

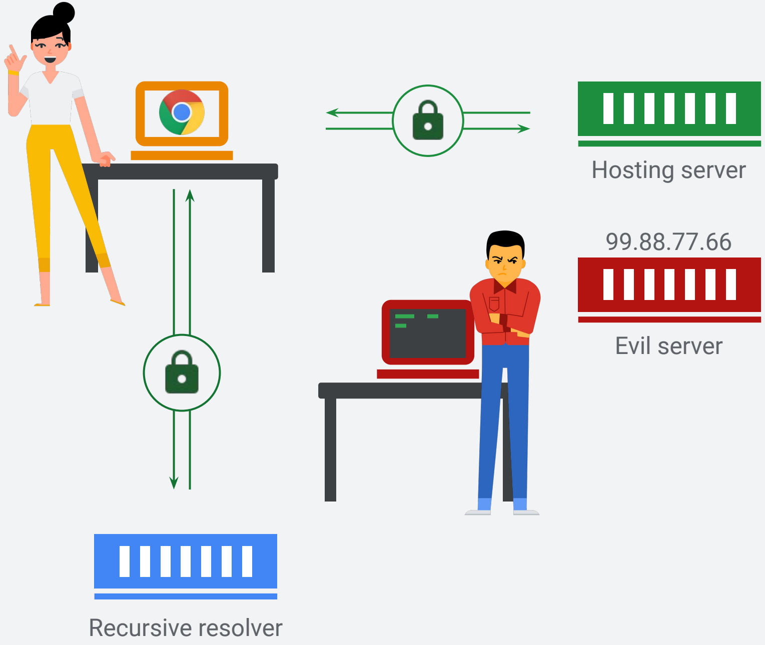
# Secure DNS

DNS-over-HTTPS makes the communication between the browser and the recursive resolver **secure** by conducting the exchange over a secure protocol, i.e HTTPS

HTTPS is designed to provide confidentiality, authentication, and integrity.

Bad actors can no longer:

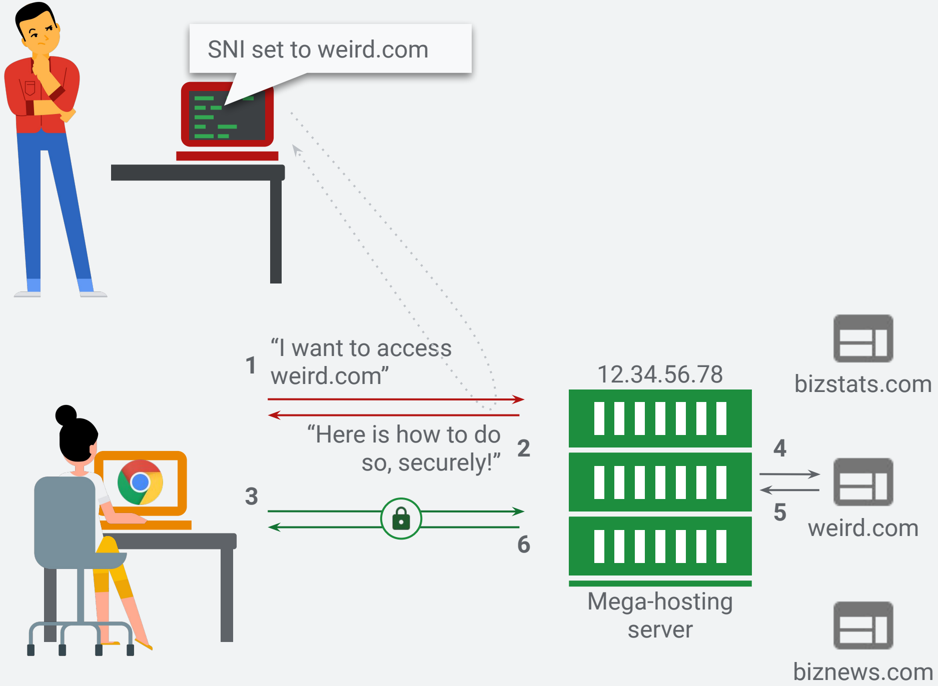
- Observe the DNS requests
- Fool the browser about the resolver’s identity
- Manipulate the response from the resolver.



# Beyond Securing DNS

DNS-over-HTTPS is **just one step** in the right direction. It's not meant to address all the privacy and security problems with Internet protocols.

For instance, **S**erver **N**ame **I**ndication* (steps 1 & 2) is **another source of privacy leak** that occurs when multiple websites are served behind a single IP address. Currently, the **desired server name** is **sent in the clear**, prior to establishing an encrypted channel.



# Complications

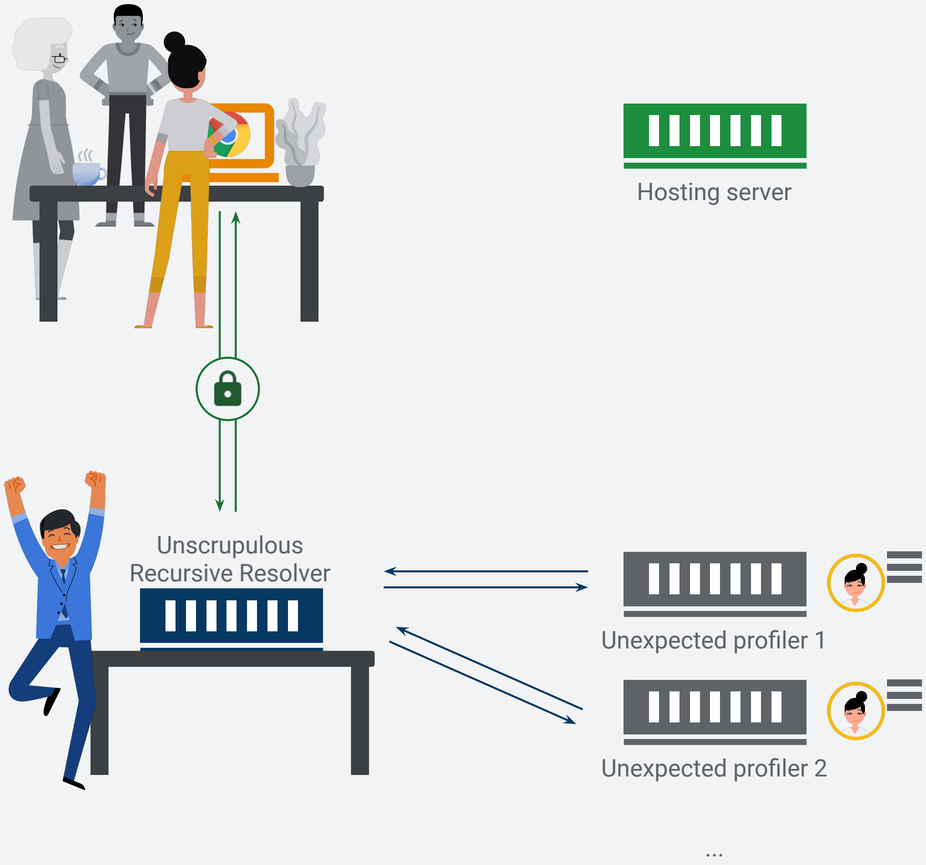
# Yeah, secure communication!
# … but to whom?

# Unscrupulous providers

**Securing** the communication channel between the browser and the recursive resolver is good but not necessarily enough.

Indeed, your browser could still be talking to a resolver operated by an **unscrupulous provider** who could sell your browsing history to third parties.

Dissociating Good from Evil is not solvable via a change of protocol.

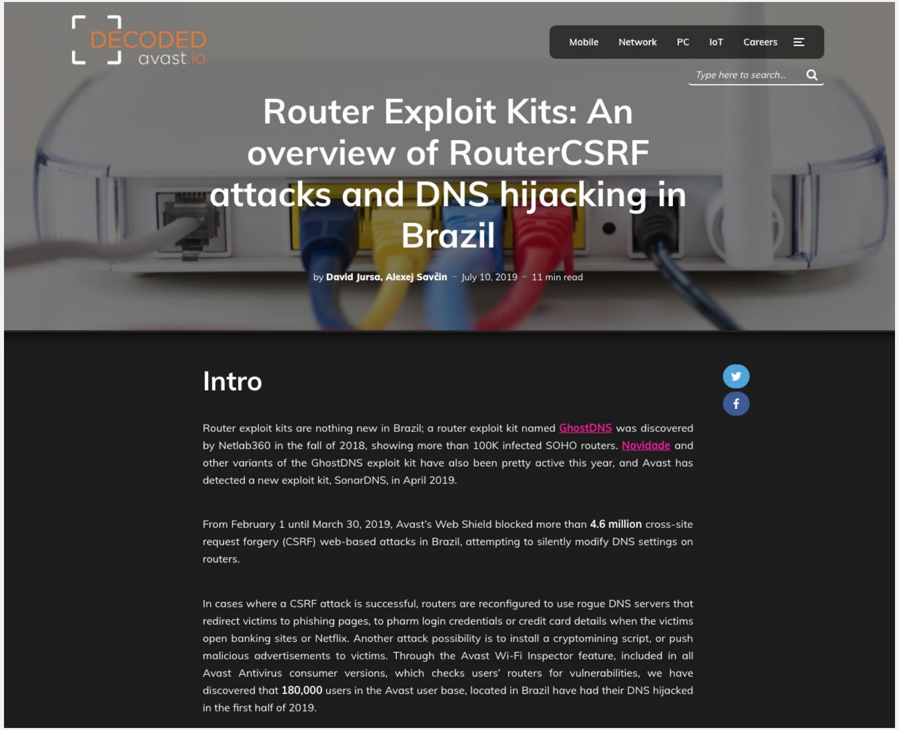


# Hacked Home Routers

Consumer grade routers have poor security track records. They also rarely receive software updates.

This makes them very appealing to bad actors. In particular, security flaws that allow to change a router's DNS settings are actively exploited to launch large scale phishing attacks, malvertising campaigns, and so on.

So, for us it means that the DNS settings advertised by home routers shouldn't be taken at face value.



## Router Exploit Kits: An overview of RouterCSRF attacks and DNS hijacking in Brazil

by **David Jursa, Alexej Savčin** – July 10, 2019 – 11 min read

### Intro

Router exploit kits are nothing new in Brazil; a router exploit kit named GhostDNS was discovered by Netlab360 in the fall of 2018, showing more than 100K infected SOHO routers. Novidade and other variants of the GhostDNS exploit kit have also been pretty active this year, and Avast has detected a new exploit kit, SonarDNS, in April 2019.

From February 1 until March 30, 2019, Avast's Web Shield blocked more than **4.6 million** cross-site request forgery (CSRF) web-based attacks in Brazil, attempting to silently modify DNS settings on routers.

In cases where a CSRF attack is successful, routers are reconfigured to use rogue DNS servers that redirect victims to phishing pages, to pharm login credentials or credit card details when the victims open banking sites or Netflix. Another attack possibility is to install a cryptomining script, or push malicious advertisements to victims. Through the Avast Wi-Fi Inspector feature, included in all Avast Antivirus consumer versions, which checks users' routers for vulnerabilities, we have discovered that **180,000** users in the Avast user base, located in Brazil have had their DNS hijacked in the first half of 2019.

Router exploit campaign in Brazil (2019)

# Historical baggage

# 35+ years of DNS = Lots of dependencies

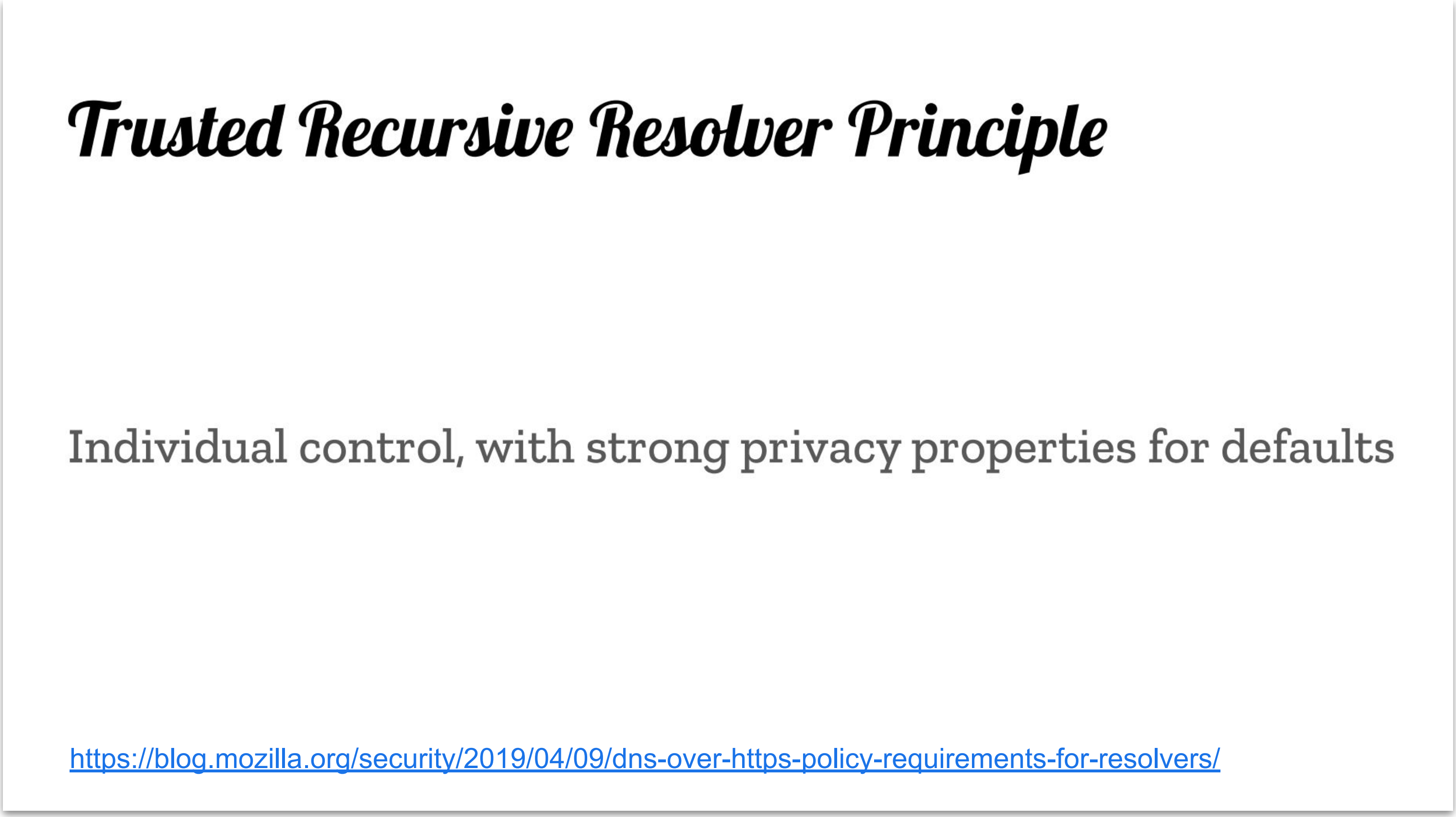
In November 1983, the Domain Name System was first described in two IETF RFCs. With 35+ years of existence, many companies have come to rely on DNS, including its insecure nature.

Some examples:

- Enterprise / Education solutions designed to prevent access to certain websites
- ISPs and router makers providing easy to setup family-safe filtering
- Security companies relying on the ability to monitor odd DNS queries for detecting new malware.
- ISPs required to block certain content by law or court order.

# Mozilla's perspective

# Takeaways

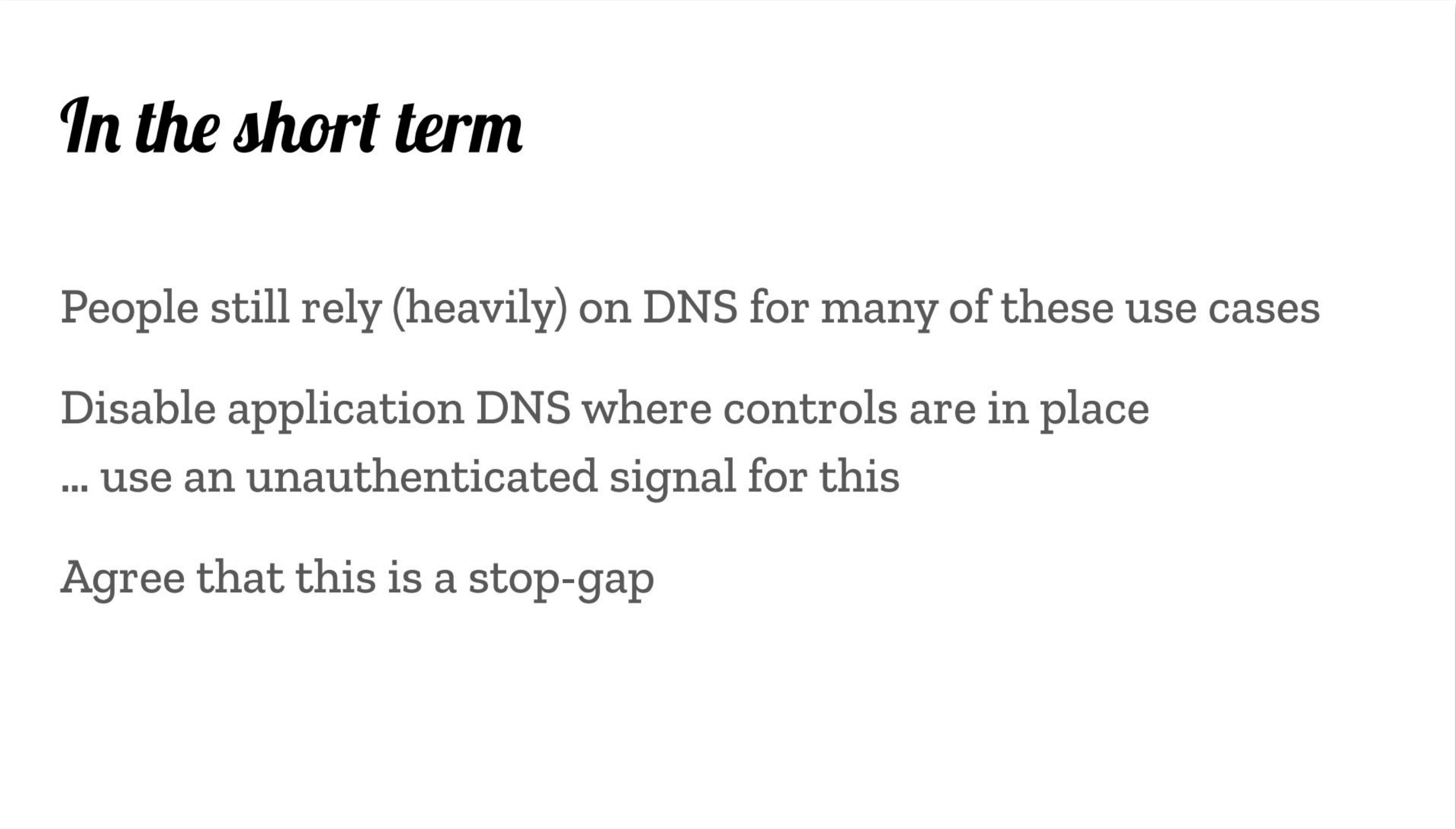


## Trusted Recursive Resolver Principle

Individual control, with strong privacy properties for defaults

https://blog.mozilla.org/security/2019/04/09/dns-over-https-policy-requirements-for-resolvers/

# Takeaways



## In the short term

People still rely (heavily) on DNS for many of these use cases

Disable application DNS where controls are in place
… use an unauthenticated signal for this

Agree that this is a stop-gap

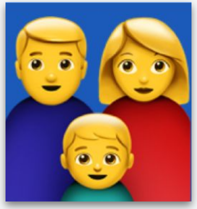
# Chrome's perspective

# Family-safe Internet

Don’t surprise our users, avoid breaking expectations.

**How:**

- Don’t force a different DNS provider (opt-in)
- Lock down / adjust UX if parental controls are enabled.
- Maintain the same experience if upgrading from a provider’s insecure DNS to that provider’s secure DNS.

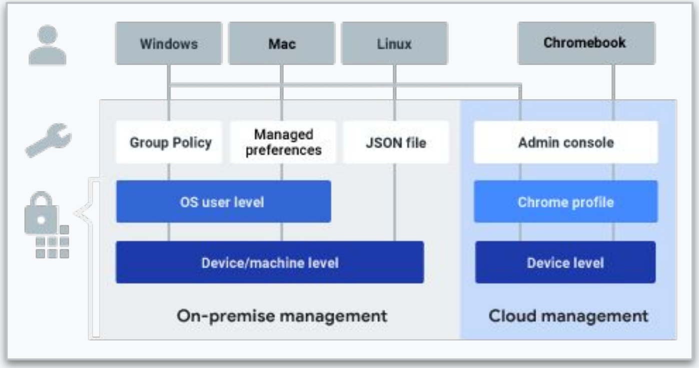


# Education/Enterprise

Continue to support Education and Enterprise use cases.

**How:** allow administrators to disable or configure DNS-over-HTTPS as they wish via **Chrome policies**.

**BYOD situations: out-of-scope**. OS specific solutions exists, e.g. Android Enterprise.



Options for enforcing Chrome policies

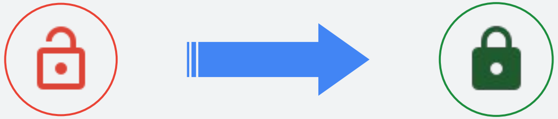
# Auto-upgrade

Targeting: Chrome 78

For users who are already using a DNS provider known to also support DoH, we will transparently auto-upgrade them to that provider's DoH service.

Requirements:

- Privacy, security stance and practice that are aligned with the motivation of DoH.
- Equivalence of service between Plain DNS and DoH, e.g. parental control, performance, availability.

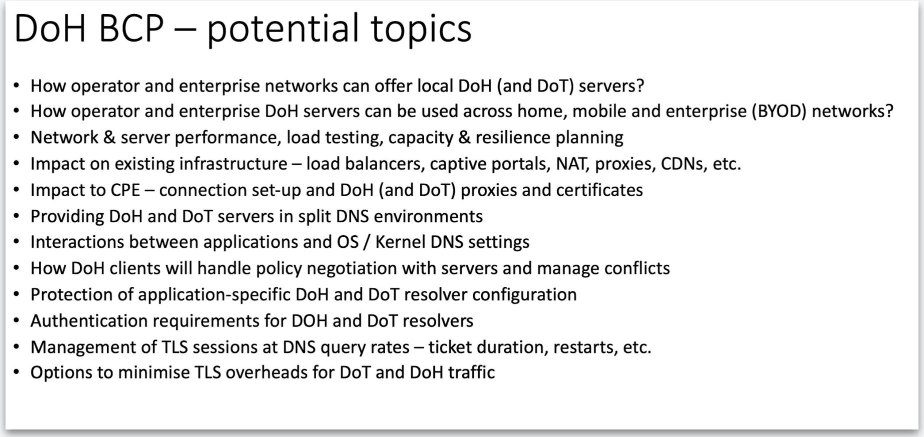


# DoH BCP

# Best Current Practices?

Presented by British Telecom

- DoH presents new challenges for operators (e.g. ISPs)
- Can ADD work on a BCP rfc?



## DoH BCP – potential topics

- How operator and enterprise networks can offer local DoH (and DoT) servers?
- How operator and enterprise DoH servers can be used across home, mobile and enterprise (BYOD) networks?
- Network & server performance, load testing, capacity & resilience planning
- Impact on existing infrastructure – load balancers, captive portals, NAT, proxies, CDNs, etc.
- Impact to CPE – connection set-up and DoH (and DoT) proxies and certificates
- Providing DoH and DoT servers in split DNS environments
- Interactions between applications and OS / Kernel DNS settings
- How DoH clients will handle policy negotiation with servers and manage conflicts
- Protection of application-specific DoH and DoT resolver configuration
- Authentication requirements for DOH and DoT resolvers
- Management of TLS sessions at DNS query rates – ticket duration, restarts, etc.
- Options to minimise TLS overheads for DoT and DoH traffic

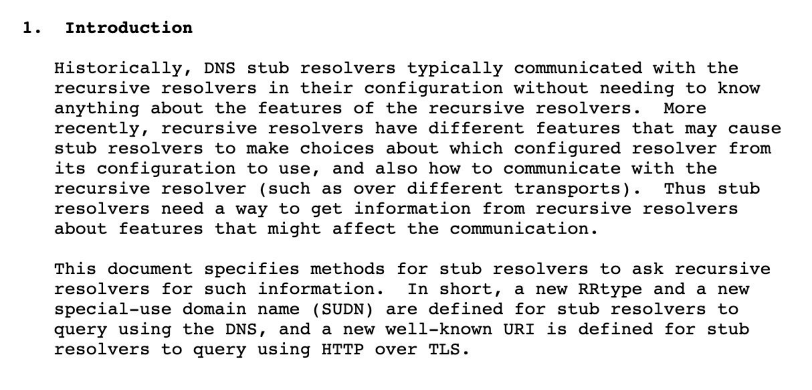
# DNS Resolver Information

# DNS Resolver Information

Allow stub resolvers to obtain more  information about a Recursive Resolver.

3 interesting suggestions at the session:

- Expose more info about the chain of resolvers involved
    - Which resolver a home router  is ultimately using?
- Expose info about who is providing the service (name of ISP)
- Which filtering features are enabled?
    - Parental control
    - Security
    - ...



```
1.  Introduction

   Historically, DNS stub resolvers typically communicated with the
   recursive resolvers in their configuration without needing to know
   anything about the features of the recursive resolvers.  More
   recently, recursive resolvers have different features that may cause
   stub resolvers to make choices about which configured resolver from
   its configuration to use, and also how to communicate with the
   recursive resolver (such as over different transports).  Thus stub
   resolvers need a way to get information from recursive resolvers
   about features that might affect the communication.

   This document specifies methods for stub resolvers to ask recursive
   resolvers for such information.  In short, a new RRtype and a new
   special-use domain name (SUDN) are defined for stub resolvers to
   query using the DNS, and a new well-known URI is defined for stub
   resolvers to query using HTTP over TLS.
```

https://tools.ietf.org/html/draft-ietf-dnsop-resolver-information-00

# Thanks

🐦 @kenjibaheux