Encrypted DNS → Privacy?
A Traffic Analysis Perspective

Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, Carmela Troncoso

NDSS, 25 February 2020
Encrypted DNS —> Privacy?

Can encrypting DNS protect users from traffic-analysis based monitoring and censoring?

We conducted a number of experiments that show that:

• Monitoring and censorship are feasible even when DNS is encrypted.
• Current proposed EDNS0-based countermeasures are not sufficient to prevent traffic analysis attacks.
The Past

Client

Query: google.com?

Response: 172.217.168.4

Recursive Resolver

Name Servers

google.com?
google.com?
google.com?

Destination Host

172.217.168.4

HTTP requests and responses
The Past

Query: google.com?

Response: 172.217.168.4

Encrypted

HTTP requests and responses

Destination Host

172.217.168.4
The Past

Client

Query: google.com?

Response: 172.217.168.4

Recursive Resolver

Name Servers

Destination Host 172.217.168.4

HTTP requests and responses

Encrypted
Encrypted DNS

DNS-over-TLS (DoT)
DNS-over-HTTPS (DoH)

Client

Recursive Resolver

Name Servers

Query: google.com?

Response: 172.217.168.4

HTTP requests and responses

Encrypted

Destination Host

172.217.168.4
Encrypted DNS

Client ➔ Query: google.com? ➔ Recursive Resolver ➔ Response: 172.217.168.4 ➔ Name Servers

HTTP requests and responses

Encrypted

Destination Host 172.217.168.4
Goal: Determine webpage visited by the client from DNS-over-HTTPS traffic.
A webpage visit can have multiple DNS queries/responses associated with it, which could be a fingerprint for identification of that webpage.
Scenario

DNS-over-HTTPS traffic

Client → Adversary

Recursive Resolver

Directionality

Size

Headers

Timing
Training

1. Collect traces
2. Extract traffic features
3. Train model on features
Training

1. Collect traces
2. Extract traffic features
3. Train model on features
Our experiment setup

1. Collect traces
2. Extract traffic features
3. Train model on features
Adversary Goal 1: Monitoring

Closed World Experiment

Set of webpages visited by user

Set of webpages known to the adversary

Which particular webpage did the user visit?
Adversary Goal 1: Monitoring

Closed World Experiment

Set of webpages visited by user

Set of webpages known to the adversary

~90% Precision and Recall

1,500 pages
Adversary Goal 1: Monitoring

Open World Experiment

Set of webpages visited by user

Set of webpages monitored by adversary

Did the user visit a page in the monitored set?
Adversary Goal 1: Monitoring

Open World Experiment

Set of webpages visited by user
Set of webpages monitored by adversary

50 pages
5,000 pages

~70% Precision and Recall
Adversary Goal 2: Censorship

Censoring adversary: **Identify webpages as fast as possible**

Study the uniqueness of DoH traffic when only the first $L$ TLS records have been observed (set of 5,000 pages).
Adversary Goal 2: Censorship

Censoring adversary: **Identify webpages as fast as possible**

Adversary strategy: **Block on first query?**

- 4th record usually corresponds to first DoH query.
- Blocking prevents user from loading the page.

- Could result in high collateral damage — pages with same domain name lengths are also blocked!

  - Iran: Blocking domain length = 13 blocks 97 domains in the censored website list, but also blocks ~86,000 domains in the Alexa top 1M list
Robustness of attack

Adversary’s training setup

DNS-over-HTTPS traffic

Client

Recursive Resolver

Selenium +

Visit webpage

Adversary

What happens when any of the parameters in this setup change?
Robustness of attack: Parameters

- **Time**
  (Dynamic Nature of websites)

- **Location**

- **Infrastructure**
  - Resolver
  - Client
  - Platform
Robustness of attack: Results

- Changes in scenario affect attack
- Adversary needs classifier tailored to scenario for best results
Monitoring and Censorship are feasible even when DNS traffic is encrypted.

Website fingerprinting using DNS traces requires \(~100\) times less data than traditional website fingerprinting.

Countermeasures?
EDNS0 Based Countermeasures

EDNS0: Extension mechanisms for DNS, specifies a padding option

Padding of DNS queries: We implemented the recommended padding strategy on Cloudflare’s DoH client. Pad query to multiples of 128 bytes.

1RFC7830
2RFC8467
Padding of DNS responses: Cloudflare’s resolver pads responses to multiples of 128 bytes. Recommended strategy: Pad to multiples of 468 bytes
Our experiments

- **EDNS0-128**: Cloudflare’s response padding strategy
- **EDNS0-468**: Recommended response padding strategy
- **Perfect Padding**: Keep all TLS record sizes constant
- **EDNS0-128-adblock**: User-side measure (ad-blocker usage)
- **DNS over Tor**: Cloudflare’s DNS over Tor service
Results: Countermeasure comparison

![Bar chart showing mean precision for different countermeasures: 90 for No countermeasure, 70 for EDNS0-128, 45 for EDNS0-468, 34 for Perfect padding, 7 for DNS over Tor, and 3.5 for Random classifier. There is a significance level of 0.001 between No countermeasure and the others.]
Results: DNS over Tor

Fixed cell sizes

Repacketization

Mean Precision (%)

No countermeasure 90
EDNS0-128 70
EDNS0-468 45
Perfect padding 34
DNS over Tor 7
Random classifier 3.5

0.001
Results: Overhead

Sent + received bytes (from TLS records)
DNS-over-HTTPS (DoH) vs DNS-over-TLS (DoT)

**DNS-over-TLS (DoT)**

**DNS-over-HTTPS (DoH)**

*Client* → *Recursive Resolver* → *Name Servers* → *Destination Host* → *Client*

- **Query:** `google.com?`
- **Response:** `172.217.168.4`

HTTP requests and responses are encrypted.
DNS-over-HTTPS (DoH) vs DNS-over-TLS (DoT)

We reran the classification process with DoT traffic

Using DoT leads to $\sim 40\%$ Precision and Recall (compared to $\sim 90\%$ for DoH)
DNS-over-HTTPS (DoH) vs DNS-over-TLS (DoT)

We reran the classification process with DoT traffic.

Using DoT leads to ~40% Precision and Recall (compared to ~90% for DoH).

DoT traffic looks different from DoH traffic.

Does traffic variability account for better protection in DoT?
Realistic scenarios

- Data pollution (Multi-tab browsing, background apps)
- Caching

Countermeasures

- Padding + repacketization measures — Can we achieve protection without using Tor?
Summary

• Surveillance and DNS-based censorship can occur even in the presence of encrypted DNS.
• Current proposed EDNS0 based countermeasures are not sufficient.
• Recommendation: Repacketization and padding

Code and datasets at:

https://github.com/spring-epfl/doh_traffic_analysis

Get in touch:  sandra.siby@epfl.ch  @sansib
BACKUP
Feature extraction

pcap file

24 -58 63 110 -92 -86 -55

TLS record sizes

24 -58 173 -233

Burst sizes

Uni-grams: (24), (-58)….

Bi-grams: (24, -58), (-58, 63)…

Uni-grams: (24), (-58)…

Bi-grams: (24, -58), (-58, 173)…

Counts
Adversary Goal 2: Censorship

Censoring adversary: Identify webpages as fast as possible

Consequences of blocking based on domain length

Censor blocking strategy

- Minimum collateral damage
- Maximum censor gain
- Most popular website
Adversary Goal 2: Censorship

Censoring adversary: *Identify webpages as fast as possible*

Adversary strategy: **High confidence guessing?**

- By 15th record (15% of trace), adversary can guess with high confidence.
- Less collateral damage.
DNS over Tor

Fixed cell sizes

• Affect size features

Repacketization

• Affect directionality features

Clusters in confusion graph?

Confusion graph of misclassified labels

Pages in a cluster are misclassified as each other
DNS-over-HTTPS (DoH) vs DNS-over-TLS (DoT)

DoT traffic looks different from DoH traffic:

• Only DNS Type A records (compared to Type A and Type AAAA in DoH)
• Even after removal of AAAA traffic, smaller number of records in DoT (more ‘bare-bones’ than DoH)
• Larger record size in DoT

*Does this traffic variability account for better protection in DoT?*