On the ineffectiveness of QUIC PADDING against website fingerprinting

Ludovic Barman, Sandra Siby, Christopher Wood, Marwan Fayed, Nick Sullivan, Carmela Troncoso

IETF PEARG, 21 March 2022
What is website fingerprinting?

Client → example.com → Encrypted channel → Destination

Adversary
What is website fingerprinting?

Client -> example.com -> Destination

Adversary

Encrypted channel

Traffic Sample

(metadata and IP addresses)
What is website fingerprinting?

Client

Destination

example.com

Encrypted channel

Adversary

Traffic Sample

Features

Pre-trained classifier

Identified webpage

Traffic characteristics used
Website Fingerprinting on QUIC

Client → example.com → Destination

Adversary

Encrypted channel
Website Fingerprinting on QUIC

Website fingerprinting on QUIC has already been studied before [1]

Conclusion: It is not harder to fingerprint QUIC as compared to TCP

Website Fingerprinting on QUIC

Website fingerprinting on QUIC has already been studied before [1].

Conclusion: It is not harder to fingerprint QUIC as compared to TCP.

QUIC RFC specifies PADDING frame [2]:

“Padding can be used … to provide protection against traffic analysis …”

Adversarial Model

AS X

AS Y

AS Z
Adversarial Model
Adversarial Model

AS X

Client

AS Y

AS Z

AS W

Destination hosts
Adversarial Model

Client → AS X → AS Y → AS W → Destination hosts

AS X

AS Y

AS W

AS Z
Adversarial Model
Adversarial Model

Identify the correct webpage among all the webpages hosted on an IP.
Dataset

- QUIC-dominant dataset (150 pages)
- ~70% QUIC on average
- Built from lists of popular domains
Process

Dataset

Traffic Sample

Defended Sample

Apply defense

Features

Pre-trained classifier

Predicted webpage

k-FP features + Random Forest / VarCNN
Unconstrained Adversary

Undefended Traffic: 96% F-Score
Unconstrained Adversary

Undefended Traffic: 96% F-Score

Random baseline: 0.67%
Unconstrained Adversary

Undefended Traffic: 96% F-Score

Size-based features are important
Unconstrained Adversary

Undefended Traffic: 96% F-Score

Hide packet-based features
(pad individual packets)

~94%
Unconstrained Adversary

Undefended Traffic: 96% F-Score

Hide packet-based features (pad individual packets) ~94%

Hide trace-based features (pad total size) ~92%
Unconstrained Adversary

**Undefended Traffic: 96% F-Score**

- Hide packet-based features (pad individual packets) ~94%
- Hide trace-based features (pad total size) ~92%

Directionality-based features become important
Unconstrained Adversary

Undefended Traffic: 96% F-Score

- Hide packet-based features (pad individual packets) ~94%
- Hide trace-based features (pad total size) ~92%
- Dummy injection

Graph: Attacker F1 score vs. Additional percentage of dummies [%]
Unconstrained Adversary

Undefended Traffic: 96% F-Score

Network defenses offer low protection with high costs

Ex: For 10% reduction in F-Score, we need >50% overhead

Hide global features (pad total size) ~94%

Hide local features (pad individual packets) ~92%
Constrained Adversary: Limited view

Client → AS X → AS Y

AS X

AS Y

AS Z

AS W

Destination hosts
Constrained Adversary: Limited view

A few large ASes can successfully run attacks.
A few large ASes can successfully run attacks.

Timings to Google resources is a low-cost fingerprint: 77.9% F-score
Constrained Adversary: Limited processing

Adversary resources
- Unconstrained
- Constrained
<table>
<thead>
<tr>
<th>Sampled NetFlow</th>
<th>Undefended (F-score)</th>
<th>Defended (F-Score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetFlow 100%</td>
<td>90.5</td>
<td>53.1</td>
</tr>
<tr>
<td>NetFlow 10%</td>
<td>66.4</td>
<td>33.1</td>
</tr>
<tr>
<td>NetFlow 1%</td>
<td>41.7</td>
<td>21.6</td>
</tr>
<tr>
<td>NetFlow 0.1%</td>
<td>16.8</td>
<td>8.6</td>
</tr>
</tbody>
</table>
## Constrained Adversary: Limited processing

<table>
<thead>
<tr>
<th>Sampled NetFlow</th>
<th>Undefended (F-score)</th>
<th>Defended (F-Score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetFlow 100%</td>
<td>90.5</td>
<td>53.1</td>
</tr>
<tr>
<td>NetFlow 10%</td>
<td>66.4</td>
<td>33.1</td>
</tr>
<tr>
<td>NetFlow 1%</td>
<td>41.7</td>
<td>21.6</td>
</tr>
<tr>
<td>NetFlow 0.1%</td>
<td>16.8</td>
<td>8.6</td>
</tr>
</tbody>
</table>

Random baseline: 0.67%
### Constrained Adversary: Limited processing

<table>
<thead>
<tr>
<th>Sampled NetFlow</th>
<th>Undefended (F-score)</th>
<th>Defended (F-Score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetFlow 100%</td>
<td>90.5</td>
<td>53.1</td>
</tr>
<tr>
<td>NetFlow 10%</td>
<td>66.4</td>
<td>33.1</td>
</tr>
<tr>
<td>NetFlow 1%</td>
<td>41.7</td>
<td>21.6</td>
</tr>
<tr>
<td>NetFlow 0.1%</td>
<td>16.8</td>
<td>8.6</td>
</tr>
</tbody>
</table>

Most of the privacy gain comes from the sampling process than the defense.
Network layer defenses cannot efficiently hide global features without application layer information.
Network layer → Application layer
Analysing web pages

Visited site (first party)

Example from img.example.com (first party)

Ad from tracker.com (third party)
Analysing web pages

18% of pages have < 20% first party resources

24% of pages have > 50% Google resources
Analysing web pages

18% of pages have < 20% first party resources

24% of pages have > 50% Google resources

Third parties contribute a large proportion of resources

All parties must participate in the protection of resources
Application layer defenses

- Packet-based and Trace-based padding

Padding is, once again, ineffective

Best case: reduces F-Score by 16% with total cost of ~8MB per resource
Application layer defenses

- Packet-based and Trace-based padding
  Padding is, once again, ineffective
  Best case: reduces F-Score by 16% with total cost of ~8MB per resource

- Dummy Injection
  Injecting dummies is more effective, but comes with deployment complexity.
  Example: Injecting 5 dummies on average reduces F-Score by 39% with total cost of ~137kB per resource
Summary

- Application-agnostic network layer defenses based on PADDING are inadequate because they fail to hide global features.

- Application-layer defenses are more effective but suffer from deployment challenges:
  - Coordination between parties
  - Developer practices
  - Client experience


Get in touch: sandra.siby@epfl.ch @sansib
Backup
Application layer defenses: Padding total sizes

- Number of sizes $N$:
  - $10^{-1}$
  - $10^{-2}$
  - $10^{-5}$

- Cost/subrequest [kB]:
  - $10^{-1}$
  - $10^{2}$
  - $10^{5}$

- Attacker F1 score:
  - 0.8
  - 0.9

- Size buckets

- Lines in the graph:
  - "incoming"
  - "outgoing"
Application layer defenses: Injecting dummies

![Graph showing the relationship between the number of dummy requests and cost per page. The x-axis represents the number of dummy requests, ranging from 0 to 30. The y-axis represents the cost per page, ranging from 0 to 6 MB. The graph plots two lines: one for cost per page and one for the Attacker F1 score. Popular requests are indicated by an arrow.]