Zero-delay Lightweight Defenses against Website Fingerprinting

Jiajun GONG, Tao Wang
Website Fingerprinting
Website Fingerprinting (WF)

Network traffic  
outgoing  incoming

Time/volume information

fingerprint

Information leakage

WF attackers: ISP, someone under the same network
Website Fingerprinting

1. Collect data
2. Train a classifier
3. Predict

- kNN [Wang et al., 2014]
- CUMUL [Panchenko et al., 2016]
- kFP [Jamie Hayes and George Danezis, 2016]
- DF [Sirinam et al., 2018]

> 90% recall

Threat to privacy!
Defense

• WTF-PAD [Juarez et al. 2016]

• Tamaraw [Cai et al. 2014]
Evaluation of a defense

- Privacy
- Overhead:

\[
data \text{ overhead} = \frac{\# \text{ dummy packets}}{\# \text{ real packets}}\]

- cost more bandwidth

\[
time \text{ overhead} = \frac{t_{\text{new}} - t_{\text{old}}}{t_{\text{old}}}\]

- causing delay

Browsing experience
Defense

Question: Better defense?

We proposed two zero-delay lightweight defenses: FRONT and GLUE.

- 0% time overhead
- Little data overhead
• General Idea

Original traffic

Client dummy

Server dummy

Obfuscated traffic

Timeline
How to schedule these dummy packets?

Intuition 1: Obfuscating feature-rich trace fronts

Why Rayleigh distribution?

$$Pr(0 < t \leq w) = 40\%$$
Set parameters: $N, W_{\text{min}}, W_{\text{max}}$

Sample a number of dummy packets

$n \propto \text{Uniform}(1, N)$

Decide the shape of distribution

$w \propto \text{Uniform}(W_{\text{min}}, W_{\text{max}})$

Sample $n$ timestamps

Intuition 2: Trace-to-trace randomness
Experiment Setup

Dataset: 100 x 100 + 10000

Monitored non-monitored  90% training, 10% testing

Attacker’s goal:

To identify whether the client is visiting a monitored page

and which monitored webpage?
Experiment Result

• Compared with WTF-PAD:

~ 33% data overhead, 0% time overhead

F1 score = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}
Experiment Result

• Compared with Tamaraw:

- F1 score
  - kNN: 0.2
  - CUMUL: 0.4
  - kFP: 0.6
  - DF: 0.8

- kNN: ~5 times

- Overhead (%)
  - Data overhead: Tamaraw 78, FRONT 49
  - Time overhead: Tamaraw 163, FRONT 0

- Attack:
  - kNN
  - CUMUL
  - kFP
  - DF
GLUE

Intuition:

GLUE

• Cover the first loading with FRONT
• “Glue” all the visits with glue traces
  ◦ fake loading, obtained by storing the history of some webpages loaded before

• Maximum duration of a glue trace: $d_{\text{max}} \propto \text{Uniform}(t_{\text{min}}, t_{\text{max}})$
Evaluation

Scenario 1: knowing $\ell$

- Randomly generated 618 ~ 4500 $\ell$-traces ($\ell$=2~16)
- Undefended dataset:
  
  82% ~ 96% recall and precision (92% split accuracy)

- GLUE dataset:
  
  4% ~ 54% recall and 4% ~ 20% precision
Evaluation
Scenario 1: without knowing $\ell$ (more realistic)

- Undefended dataset:
  45% ~ 75% recall and 41% ~ 77% precision

- GLUE dataset:
  3% ~ 46% recall and 1% ~ 16% precision
Overhead of GLUE

• Time overhead 0%.

• Suppose:

  • mean dwell time $d_G$, mean duration of tail $d_L$

22-44% data overhead
Summary

- Proposed two lightweight zero-delay defenses:
  - **FRONT** injects dummy packets in a traditional way
    - Obfuscating the trace fronts
    - Trace-to-trace randomness
  - **GLUE** explores a new direction for designing a defense
    - Forces the attacker to solve the split problem
• Source code

https://github.com/websitefingerprinting/WebsiteFingerprinting/

• Contact info:

jgongac@cse.ust.hk

Thanks for listening!