Website Fingerprinting

Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier

Dominik Herrmann, Hannes Federrath
University of Regensburg, Germany

Rolf Wendolsky
JonDos GmbH
Motivation – To Whom It May Concern

- Various **Privacy Enhancing Technologies (PET)** offer protection against eavesdropping
  - SSH/SSL tunnels and VPNs
  - multi-hop anonymisation services

- Users want protection against malicious ISPs and other users
- Criminals want to hide their activities from the authorities
Attack Scenario

e.g. VPN, OpenSSH tunnel, Tor, ...

e.g. ISP, local admin, authorities, ...

client

tunnel endpoint

destination webservers

encrypted traffic

attacker
Overview of Our Fingerprinting Attack

- Attacker wants to learn URLs of websites that are requested over an encrypted tunnel by the victim.

- **Website Fingerprints:** Attack exploits characteristic structure of websites.

- **Attacker:** passive, local, external observer

**PROCEDURE**

- Set up a database with traffic profiles of all websites of interest (training phase)

- Compare observed traffic with all profiles from database to predict likely candidates
Overview of Our Fingerprinting Attack

- Attacker wants to learn URLs of websites that are requested over an encrypted tunnel by the victim.

- **Website Fingerprints:** An attack exploits characteristic structure of websites.

- **Attacker:** passive, local, external observer

**PROCEDURE**

- Set up a database with traffic profiles of all websites of interest (training phase)

- Compare observed traffic with profiles in database to predict likely candidates
Overview of Our Fingerprinting Attack

- Attacker wants to learn URLs of websites that are requested over an encrypted tunnel by the victim.

- **Website Fingerprints:** Attack exploits characteristic structure of websites.

- **Attacker:** passive, local, external observer

**PROCEDURE**

- Set up a database with traffic profiles of all websites of interest (training phase)

- Compare observed traffic with all profiles from database to predict likely candidates
Previous works concentrate on OpenSSH and two well-known fingerprinting techniques.

Operating on file sizes:

- Sun et al. (2002)

  but: file sizes cannot be observed in encrypted tunnels!

Operating on IP packet sizes:

- Bissias et al. (2005): identify only 20% of sites
- Liberatore & Levine (2006): identify up to 73% of sites using Jaccard coefficient and Naïve-Bayes classifier
Focus of Our Paper

Operating on file sizes:
- Sun et al. (2002)
- but: file sizes cannot be observed in encrypted tunnels!

Operating on IP packet sizes:
- Bissias et al. (2005): identify only 20% of sites
- Liberatore & Levine (2006): identify up to 73% of sites using Jaccard coefficient and Naïve-Bayes classifier

Can we improve accuracy?

What about other PETs?

Does it work in practice?
Novel Fingerprinting Technique

Motivation and Scenario

Evaluation

Addressing Real-World Issues
Modeling Website Fingerprinting as Supervised Learning Problem

class = URLs
instance = observed IP packets
attribute = packet size
attribute value = packet size frequency

Example:

- class: www.yahoo.com
- some instance: -160, 1500, 468, -52, 1500, 1500, -52, 1500
- set representation: (-160, -52, 468, 1500)
- vector representation: (1, 2, 1, 4)
Review of Existing Fingerprinting Techniques

- **Jaccard Coefficient**
  - $\text{sim}(A, B) = \frac{|A \cap B|}{|A \cup B|}; \text{sim}(A, B) \in [0;1]$
  - Operates on set representation of instances
  - Poor accuracy for padded packets

- **Naïve Bayes Classifier**
  - Estimates probability density function for each packet size
  - Increased accuracy with *Kernel Density Estimation (KDE)*
  - Overfitting if only similar training instances are available
Our Fingerprinting Technique: Multinomial Naïve Bayes (MNB) Classifier

- Popular classifier in **text mining** domain (spam detection)
- We believe that Website Fingerprinting is a similar problem.

- Operates on packet size frequency distribution

- **Idea:** the more often the most important packet sizes of the test instance $i$ appear in traces belonging to class $c$, the more likely does instance $i$ belong to class $c$

- Low computational complexity
Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies
Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies

- **IDF transformation**
  scale down frequencies of terms that are not characteristic for a class (inverse document frequency)
Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies.
Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies

- **IDF transformation**
  scale down frequencies of terms that are not characteristic for a class (inverse document frequency)

- **Cosine normalisation**
  normalise attribute vectors to uniform length (division by Euclidean length of each vector)
Agenda

Motivation and Scenario

Novel Fingerprinting Technique

Evaluation

Addressing Real-World Issues
Data Collection Methodology

- We obtained real-world traffic dumps from 775 popular domains
- Automated Firefox to download each site multiple times
- Recorded packet size and direction with `tcpdump`
- 300,000 traffic dumps for various PET systems within two months

Dataset will be available at our site for future research: [http://www-sec.uni-r.de/website-fingerprinting/](http://www-sec.uni-r.de/website-fingerprinting/)
Best Accuracy for TF Transformation and Normalisation

Normalisation makes classifier operate on relative packet frequencies.

![Graph showing accuracy comparison between raw and normalised data with different transformations. The x-axis represents the type of transformation (TF, IDF, TF-IDF, none) and the y-axis represents accuracy. The graph shows that normalisation with TF-IDF transformation yields the highest accuracy.]
Multinomial Naïve Bayes with $TF$ and $normalisation$:

- Already 90% accuracy for 1 training instance; 94% for 4 instances
- No substantial increase for more than 4 training instances

- Fingerprints built from frequency distribution of IP packet sizes are very robust against changes to contents of sites.
- Accuracy with old fingerprints decreases rather slowly: still over 90% after 17 days

Cannot directly compare these results with previous work!
Benchmarking Existing Website Fingerprinting Techniques with Our Sample

OpenSSH, 4 training and 4 test instances, \( \text{delta}_t = 6 \text{ days} \)

- **highest accuracy**: MNB with TF+normalisation

- **Naïve Bayes** really needs absolute packet frequencies

- can reproduce good accuracy of Jaccard coefficient from previous work

\[ \text{NB with KDE and Jaccard perform better than in previous studies; i.e. results not comparable across samples!} \]
Attacking Popular PETs Using the MNB Classifier

**SINGLE HOP SYSTEMS**
- Stunnel
- OpenSSH
- Cisco IPSec VPN
- OpenVPN

**MULTI HOP SYSTEMS**
- JonDonym (*aka* JAP/AN.ON)
- Tor
## Attacking Popular PETs Using the MNB Classifier

<table>
<thead>
<tr>
<th>SINGLE HOP SYSTEMS</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stunnel</td>
<td>97.6%</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>96.7%</td>
</tr>
<tr>
<td>Cisco IPSec VPN</td>
<td>96.2%</td>
</tr>
<tr>
<td>OpenVPN</td>
<td>94.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MULTI HOP SYSTEMS</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>JonDonym (aka JAP/AN.ON)</td>
<td>20.0%</td>
</tr>
<tr>
<td>Tor</td>
<td>3.0%</td>
</tr>
</tbody>
</table>
Attacking Popular PETs Using the MNB Classifier

### SINGLE HOP SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stunnel</td>
<td>97.6%</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>96.7%</td>
</tr>
<tr>
<td>Cisco IPSec VPN</td>
<td>96.2%</td>
</tr>
<tr>
<td>OpenVPN</td>
<td>94.9%</td>
</tr>
</tbody>
</table>

### MULTI HOP SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>JonDonym (aka JAP/AN.ON)</td>
<td>20.0%</td>
</tr>
<tr>
<td>Tor</td>
<td>3.0%</td>
</tr>
</tbody>
</table>

Still way better than random guessing; $p = \frac{1}{775} = 0.58\%$
## Attacking Popular PETs Using the MNB Classifier

<table>
<thead>
<tr>
<th>SINGLE HOP SYSTEMS</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stunnel</td>
<td>97.6%</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>96.7%</td>
</tr>
<tr>
<td>Cisco IPSec VPN</td>
<td>96.2%</td>
</tr>
<tr>
<td>OpenVPN</td>
<td>94.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MULTI HOP SYSTEMS</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>JonDonym (aka JAP/AN.ON)</td>
<td>20.0%</td>
</tr>
<tr>
<td>Tor</td>
<td>3.0%</td>
</tr>
</tbody>
</table>

*with 10 guesses*
# Attacking Popular PETs Using the MNB Classifier

## SINGLE HOP SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
<th>Best Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stunnel</td>
<td>97.6%</td>
<td>TF-N</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>96.7%</td>
<td>TF-N</td>
</tr>
<tr>
<td>Cisco IPSec VPN</td>
<td>96.2%</td>
<td>TF-N</td>
</tr>
<tr>
<td>OpenVPN</td>
<td>94.9%</td>
<td>TF-N</td>
</tr>
</tbody>
</table>

## MULTI HOP SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
<th>Best Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>JonDonym (aka JAP/AN.ON)</td>
<td>20.0% with 10 guesses</td>
<td>N</td>
</tr>
<tr>
<td>Tor</td>
<td>3.0%</td>
<td>N</td>
</tr>
</tbody>
</table>
# Attacking Popular PETs Using the MNB Classifier

## SINGLE HOP SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
<th>Best Classifier</th>
<th>No. of Unique Packet Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stunnel</td>
<td>97.6%</td>
<td>TF-N</td>
<td>1605</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>96.7%</td>
<td>TF-N</td>
<td>420</td>
</tr>
<tr>
<td>Cisco IPSec VPN</td>
<td>96.2%</td>
<td>TF-N</td>
<td>108</td>
</tr>
<tr>
<td>OpenVPN</td>
<td>94.9%</td>
<td>TF-N</td>
<td>2898</td>
</tr>
</tbody>
</table>

## MULTI HOP SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
<th>Best Classifier</th>
<th>No. of Unique Packet Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>JonDonym (aka JAP/AN.ON)</td>
<td>20.0%</td>
<td>N</td>
<td>205</td>
</tr>
<tr>
<td>Tor</td>
<td>3.0%</td>
<td>47.5%</td>
<td>869</td>
</tr>
</tbody>
</table>

No correlation with accuracy!

With 10 guesses
Discussion of Results

- **OpenSSH results indicative** for all studied single-hop systems
- Low accuracies for multi-hop systems due to
  - **fixed-length packages** (e.g. Tor has cell size of 512 bytes)
  - **noise** (e.g. due to TCP retransmissions)
- We **cannot conclude** that multi-hop systems are immune against fingerprinting attacks!
- **System-specific attacks** will likely achieve higher accuracies.
Agenda

Motivation and Scenario

Novel Fingerprinting Technique

Evaluation

Addressing Real-World Issues
Results obtained using research assumptions from related studies:

- **Knowledge about victim**: attacker uses similar browser, Internet access and PET system to build fingerprints database

- **Closed-world**: classifier will never encounter traffic of a site it hasn’t been trained for

- **Browser configuration**: no caching, no prefetching, no update checks

- **Extractable profiles**: attacker can extract traffic of individual page impressions from encrypted stream
Evaluation of Two Real-World Issues with OpenSSH Dataset

ENABLING BROWSER CACHE

- Previous work suggests that fingerprinting becomes difficult once browser cache is enabled.
- Cannot reproduce this with our sample: accuracy drops by only 5%

FALSE ALARMS

- Leaving closed world scenario behind: false alarms for uninteresting sites become a problem
- If only 78 of 775 pages are considered interesting,
  - 1.5% of uninteresting instances cause a false alarm
  - 40% of instances from interesting sites are classified correctly
Areas of Future Work

- Assess utility for **forensics**:
  tune attack for recognition of a very small number of sites

- Evaluate protection of **countermeasures**:  
  e.g. *Traffic Flow Confidentiality* by Kiraly et al. (2008)

- Applicability to **Cloud Computing** protocols:  
  must pay attention to traffic profile of messages
Website Fingerprinting

- Introduced **Multinomial Naïve Bayes classifier**
- Operates on **transformed relative IP packet size frequencies**
- **Higher effectivity/efficiency** for OpenSSH than existing fingerprinting techniques (accuracy of up to 97%)
- Attack also relevant for **PETs with fixed-size messages** (with limited accuracy)
- **Browser caching** is apparently negligible