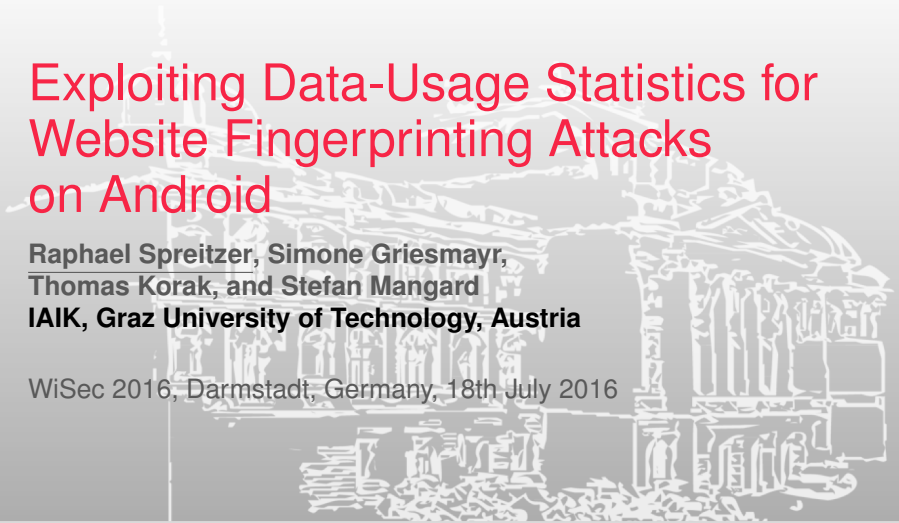
Exploiting Data-Usage Statistics for Website Fingerprinting Attacks on Android

Raphael Spreitzer, Simone Griesmayr,
Thomas Korak, and Stefan Mangard
IAIK, Graz University of Technology, Austria

WiSec 2016, Darmstadt, Germany, 18th July 2016



Contributions

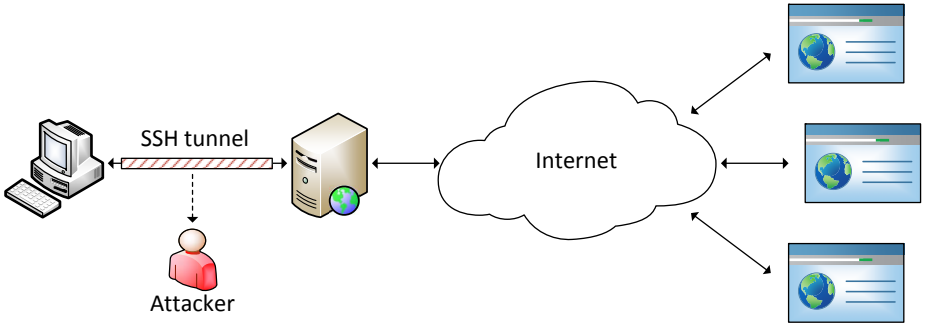
Side-channel attack to infer browsing behavior

- Unprivileged application
- Data-usage statistics
- High accuracy
- Also works when traffic is routed through Tor
- `READ_HISTORY_BOOKMARKS` does not provide protection

Website Fingerprinting

Traditional attack scenario

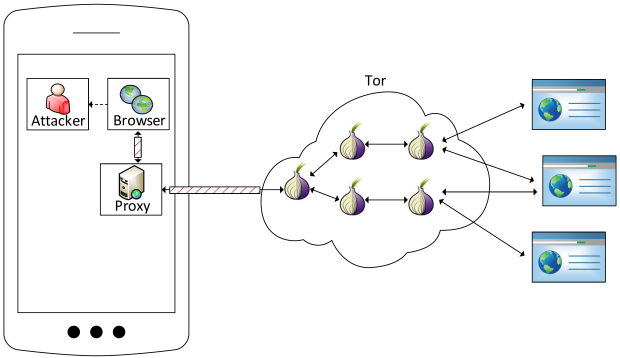
- Attacker located somewhere on the victim's network
- Traffic analysis techniques to infer browsing behavior



Website Fingerprinting

Attack scenario against smartphones

- Malicious application running in unprivileged mode
- Observe information “leaking” from browser application



Data-Usage Statistics

What is this?

- Track the amount of incoming/outgoing network traffic
- Users can stick to their data plan
- Available to all apps w/o any permission

Availability

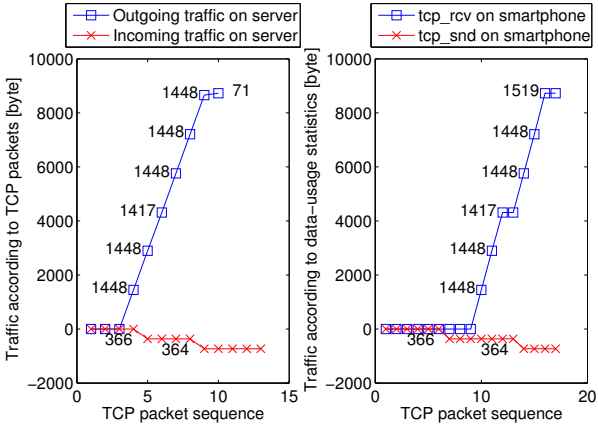
- `/proc/uid_stat/[uid]/tcp_rcv|tcp_snd`
- Android API `TrafficStats.getUidRxBytes`, `.getUidTxBytes`
- How to get `uid`?
  - `ActivityManager.getRunningAppProcesses()` (`REAL_GET_TASKS`?)
  - `PackageManager.getInstalledApplications()`

High resolution (single TCP packet lengths)

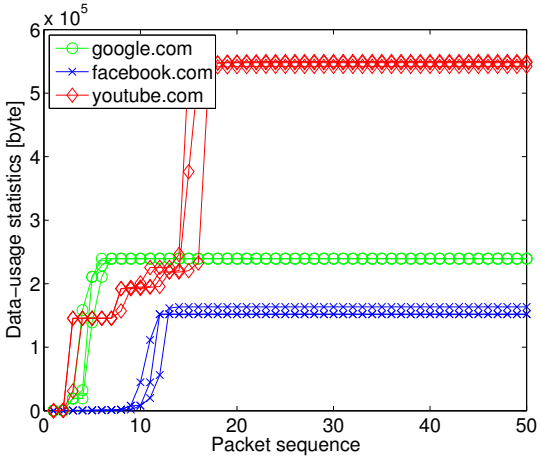
Data-Usage Statistics

Experiment

- Local server hosting a website (`tcpdump`)
- Launch website on Android (data-usage statistics)



Usage Statistics for Real Websites



Websites are distinguishable

- Stable: signatures of repeated visits to the same page are similar
- Diverse: signatures of different pages vary

Adversary Model and Attack Scenario

Adversary model

- Traditional: nw-based attacker
- Unprivileged app distributed via app market

Attack

1. Training phase (offline)
2. Attack phase (online)

Website Fingerprinting

Training phase

- Observe data-usage statistics while loading specific websites
- $\Rightarrow$ build signature database
- No “fancy” machine-learning approach
- $\Rightarrow$ no expensive training phase necessary

Website Fingerprinting

Attack phase

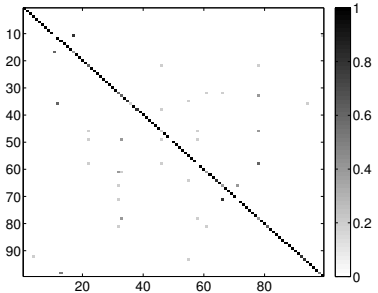
1. Distribute malicious application
2. Observe data-usage statistics for browser application
3. Infer visited website by means of signature database
   - Similarity metric for traces

$$SIM(t_1, t_2) = \frac{|t_1 \cap t_2|}{|t_1 \cup t_2|}$$

Results

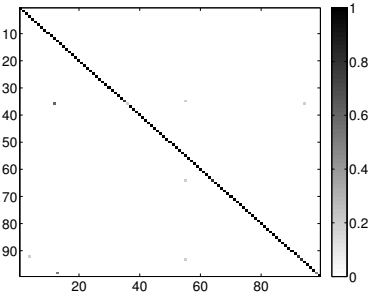
Intra-day classification rate

- 100 most popular websites globally



89% of 500 page visits

confusion of google*.* pages



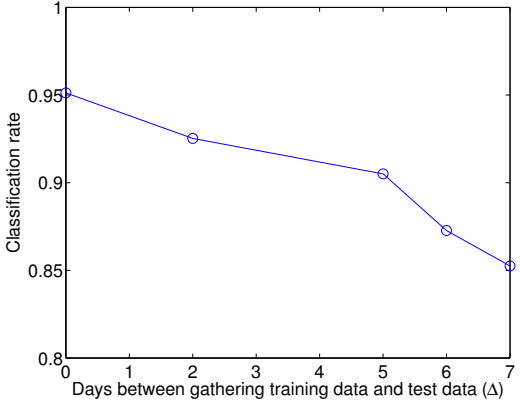
98% of 500 page visits

with google*.* pages merged

Results

Inter-day classification for Tor

- 100 most popular websites in the US



Results

Websites with the highest number of misclassifications

Δ	Website	# misclassifications
2 days	ask.com	5 times
2 days	twitch.tv	5 times
2 days	cnn.com	3 times
5 days	bbc.com	5 times
5 days	indeed.com	5 times
5 days	nytimes.com	5 times
5 days	twitch.tv	5 times
5 days	espn.go.com	4 times

Comparison

Work	Exploited information	Countermeasure	# websites	Classification rate
Ours	Client-side data-usage statistics	None	500	**97%**
Jana and Shmatikov [JS12]	Client-side memory footprint	None	100	35%
Ours	Client-side data-usage statistics	Tor	100	**95%**
Wang et al. [WCN+14]	TCP packets	Tor	100	95%
Wang and Goldberg [WG13]	TCP packets	Tor	100	91%
Cai et al. [CZJJ12]	TCP packets captured via tshark	Tor	100	84%
Panchenko et al. [PNZE11]	Client-side tcpdump	Tor	775	55%
Herrmann et al. [HWF09]	Client-side tcpdump	Tor	775	3%

Advantages

- Ease of applicability (unprivileged app vs on the wire)
- Computational performance (no training vs 608 000 CPU seconds)
- Classification rates
- No traffic noise due to other apps

Countermeasures

Against NW-based fingerprinting attacks

- Traffic morphing, HTTPOS, BuFLO, Glove
- Tor?

Client-side countermeasures

- Permission-based approaches? [ZDH+13]
  - Request permission to monitor data-usage statistics?
  - Let developers specify how statistics should be published?

$\Rightarrow$ update data-usage statistics on a more coarse-grained level

Conclusions

Fundamental weaknesses in Android

- Seemingly innocuous information
- . . . that turns out to be a serious information leak

Unprivileged app can infer browsing behavior, although

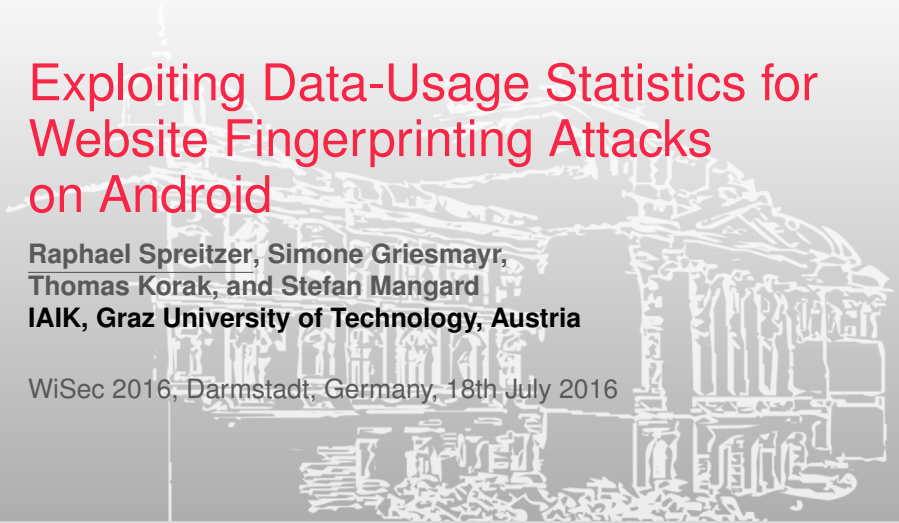
- Orweb or “private/incognito” modes do not store browsing history
- Traffic is routed through Tor
- `READ_HISTORY_BOOKMARKS` should protect this sensitive information

$\Rightarrow$ Privacy issue

Exploiting Data-Usage Statistics for Website Fingerprinting Attacks on Android

Raphael Spreitzer, Simone Griesmayr,
Thomas Korak, and Stefan Mangard
IAIK, Graz University of Technology, Austria

WiSec 2016, Darmstadt, Germany, 18th July 2016



Bibliography I

[CZJJ12] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson.
Touching from a Distance: Website Fingerprinting Attacks and Defenses.
In *Conference on Computer and Communications Security – CCS 2012*, pages 605–616. ACM, 2012.

[HWF09] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath.
Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier.
In *Cloud Computing Security Workshop – CCSW*, pages 31–42. ACM, 2009.

[JS12] Suman Jana and Vitaly Shmatikov.
Memento: Learning Secrets from Process Footprints.
In *IEEE Symposium on Security and Privacy – S&P 2012*, pages 143–157. IEEE Computer Society, 2012.

[PNZE11] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel.
Website Fingerprinting in Onion Routing Based Anonymization Networks.
In *Workshop on Privacy in the Electronic Society – WPES 2011*, pages 103–114. ACM, 2011.

Bibliography II

[WCN+14] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg.

Effective Attacks and Provable Defenses for Website Fingerprinting.

In *USENIX Security Symposium 2014*, pages 143–157. USENIX Association, 2014.

[WG13] Tao Wang and Ian Goldberg.

Improved Website Fingerprinting on Tor.

In *Workshop on Privacy in the Electronic Society – WPES 2013*, pages 201–212. ACM, 2013.

[ZDH+13] Xiao-yong Zhou, Soteris Demetriou, Dongjing He, Muhammad Naveed, Xiaorui Pan, XiaoFeng Wang, Carl A. Gunter, and Klara Nahrstedt.

Identity, Location, Disease and More: Inferring Your Secrets from Android Public Resources.

In *Conference on Computer and Communications Security – CCS 2013*, pages 1017–1028. ACM, 2013.