Side-Channel Attacks on Human Secrets

Yossi Oren, BGU

https://iss.oy.ne.ro

@yossioren

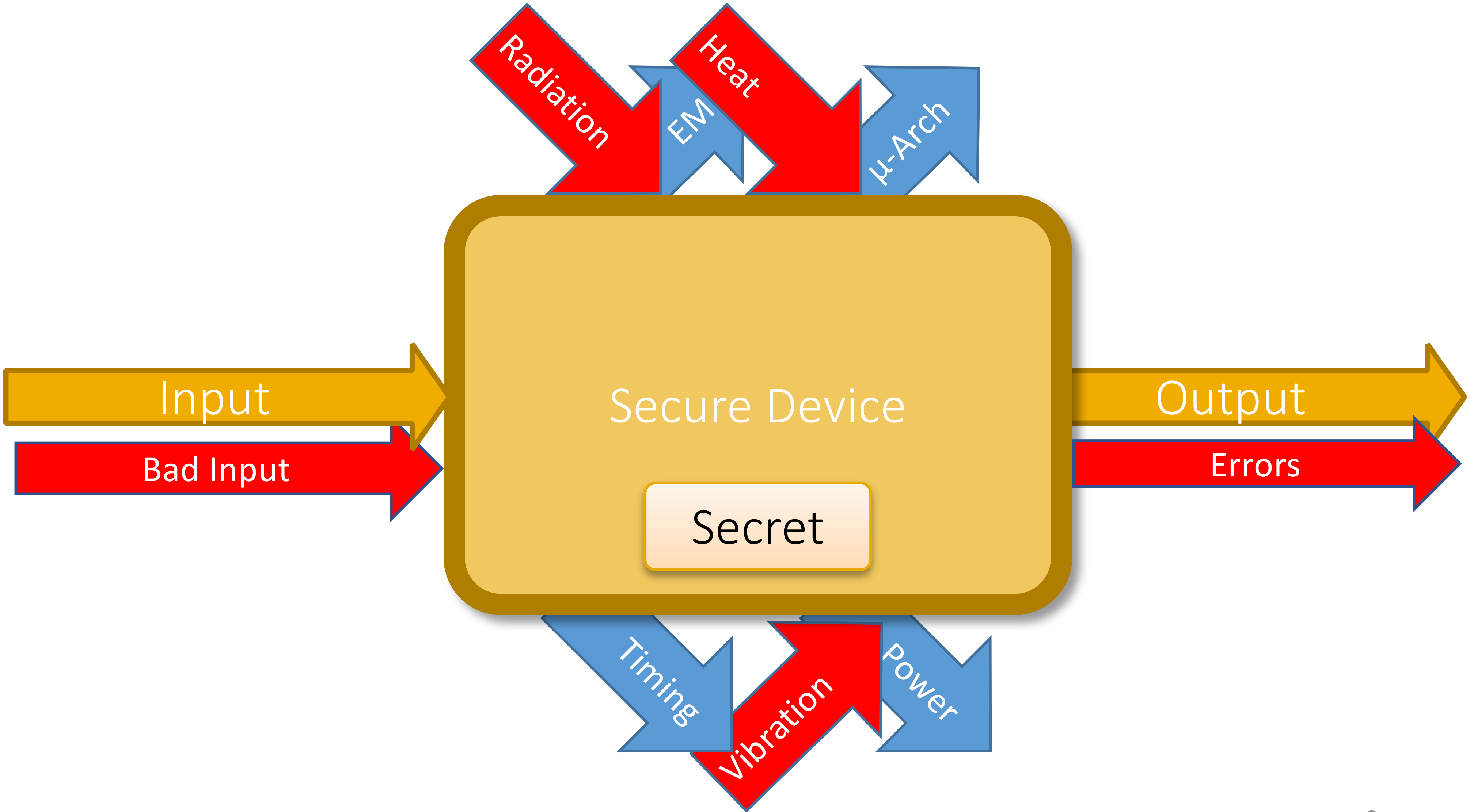
SILM Summer School, Rennes (France), July 2019

Joint work with Anatoly Shusterman, Lachlan Kang, Yosef Meltser, Yarden Haskal, Prateek Mittal and Yuval Yarom

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

https://orenlab.sise.bgu.ac.il

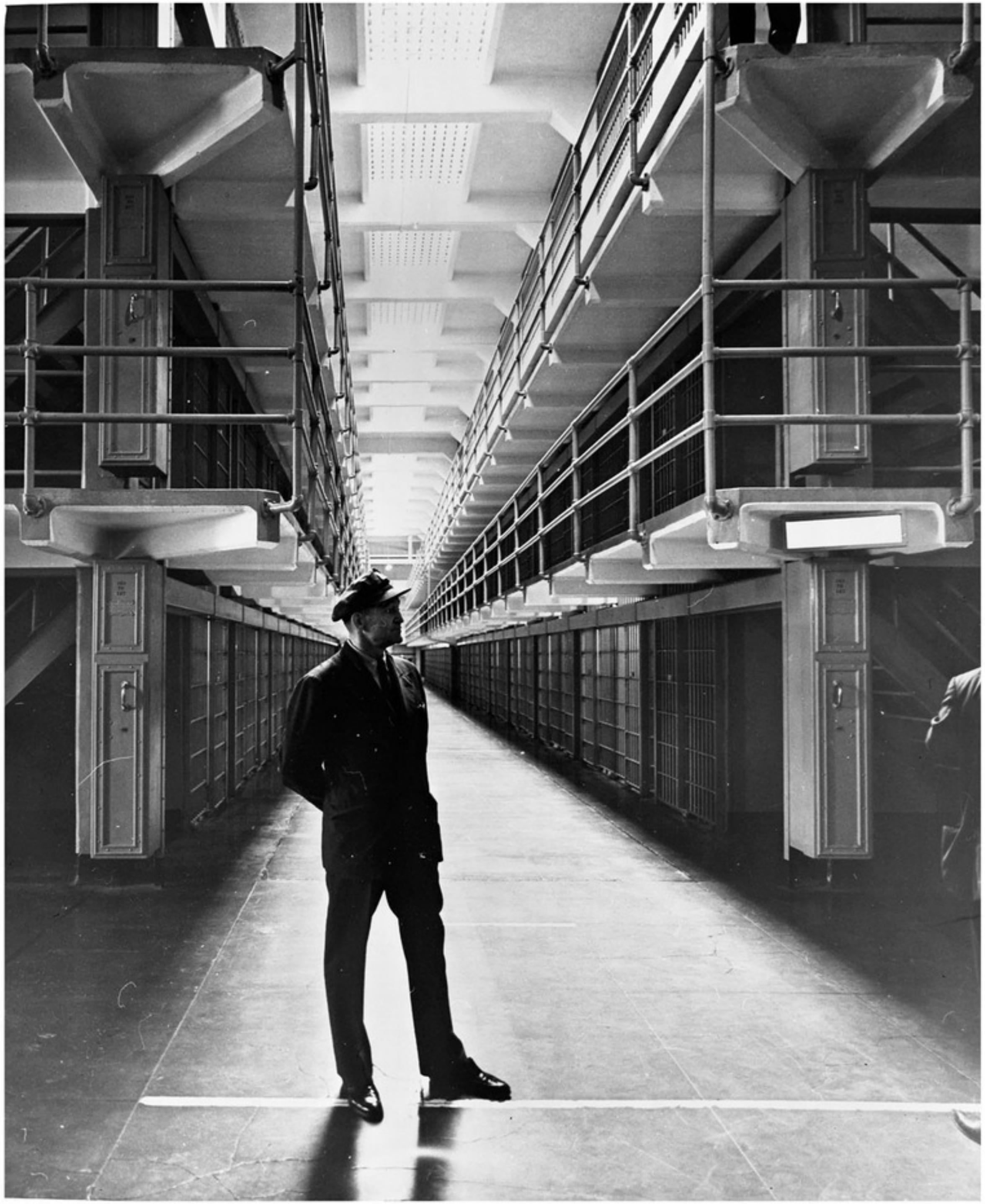
Implementation Attacks



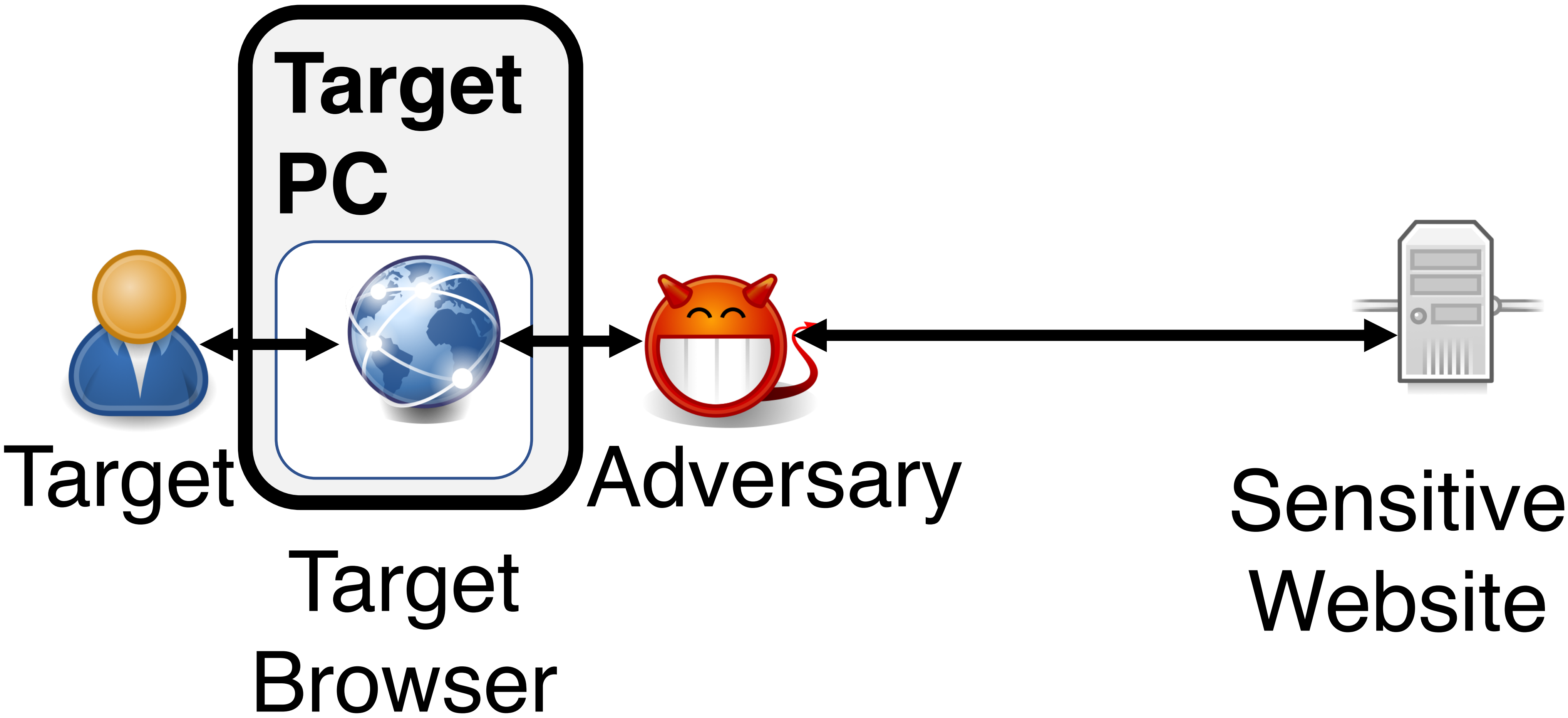
Types of Secrets

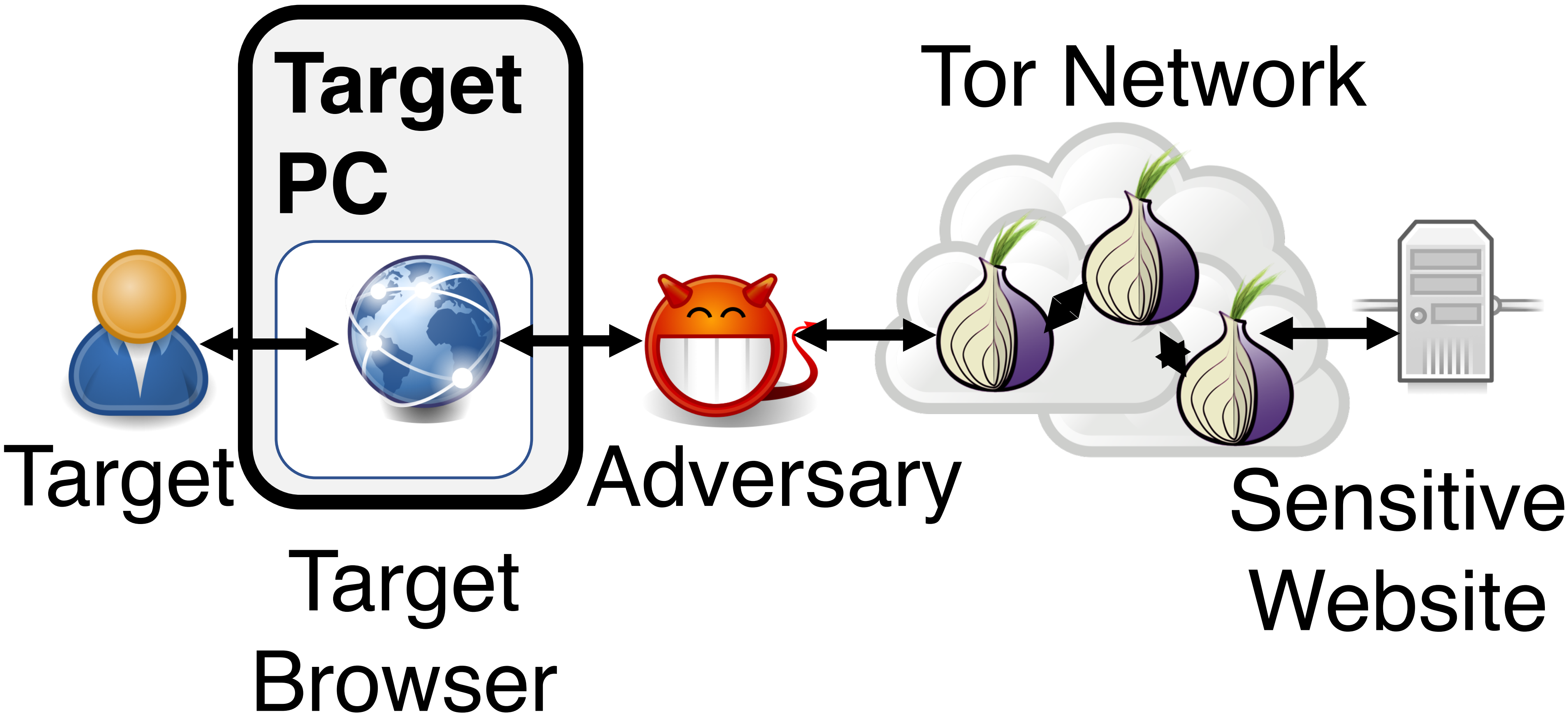
Crypto Secrets	State Secrets	Human Secrets
Short-Term Session Keys	Addresses of Sensitive Instructions	Identity
		Passwords
Long-Term Signing Keys	Inventory of Installed Vulnerable Software	Browsing History
		Images on Screen
Long-Term Decryption Keys	Random Number Generator State	Health Sensors

- What if the secret is compromised?
- How do we protect the secret from attack?

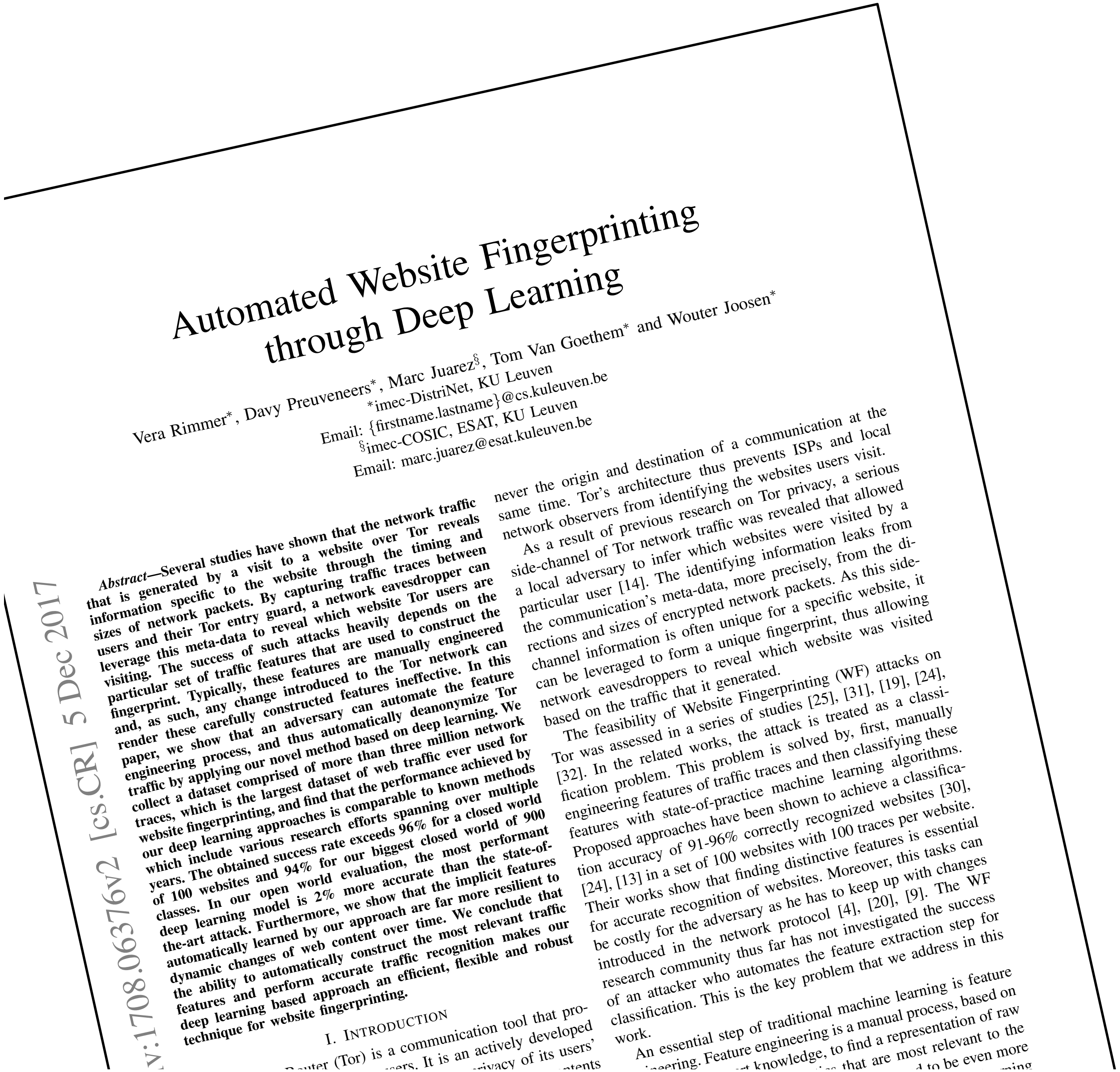


Credit: SF Public Library courtesy of Golden Gate NRA, Park Archives, Interpretive Negative Collection, GOGA-2316


Target
PC
Target
Target
Browser
Adversary
Sensitive
Website


Tor Network
Target
PC
Target
Target
Browser
Adversary
Sensitive
Website

Website Fingerprinting


v:1708.06376v2 [cs.CR] 5 Dec 2017

Automated Website Fingerprinting through Deep Learning

Vera Rimmer*, Davy Preuveneers*, Marc Juarez§, Tom Van Goethem* and Wouter Joosen*
*imec-DistriNet, KU Leuven
Email: {firstname.lastname}@cs.kuleuven.be
§imec-COSIC, ESAT, KU Leuven
Email: marc.juarez@esat.kuleuven.be

***Abstract*—Several studies have shown that the network traffic that is generated by a visit to a website over Tor reveals information specific to the website through the timing and sizes of network packets. By capturing traffic traces between users and their Tor entry guard, a network eavesdropper can leverage this meta-data to reveal which website Tor users are visiting. The success of such attacks heavily depends on the particular set of traffic features that are used to construct the fingerprint. Typically, these features are manually engineered and, as such, any change introduced to the Tor network can render these carefully constructed features ineffective. In this paper, we show that an adversary can automate the feature engineering process, and thus automatically deanonymize Tor traffic by applying our novel method based on deep learning. We collect a dataset comprised of more than three million network traces, which is the largest dataset of web traffic ever used for website fingerprinting, and find that the performance achieved by our deep learning approaches is comparable to known methods which include various research efforts spanning over multiple years. The obtained success rate exceeds 96% for a closed world of 100 websites and 94% for our biggest closed world of 900 classes. In our open world evaluation, the most performant deep learning model is 2% more accurate than the state-of-the-art attack. Furthermore, we show that the implicit features automatically learned by our approach are far more resilient to dynamic changes of web content over time. We conclude that the ability to automatically construct the most relevant traffic features and perform accurate traffic recognition makes our deep learning based approach an efficient, flexible and robust technique for website fingerprinting.**

I. Introduction

...Router (Tor) is a communication tool that pro- ...users. It is an actively developed ...rivacy of its users' ...ntents

never the origin and destination of a communication at the same time. Tor's architecture thus prevents ISPs and local network observers from identifying the websites users visit.

As a result of previous research on Tor privacy, a serious side-channel of Tor network traffic was revealed that allowed a local adversary to infer which websites were visited by a particular user [14]. The identifying information leaks from the communication's meta-data, more precisely, from the directions and sizes of encrypted network packets. As this side-channel information is often unique for a specific website, it can be leveraged to form a unique fingerprint, thus allowing network eavesdroppers to reveal which website was visited based on the traffic that it generated.

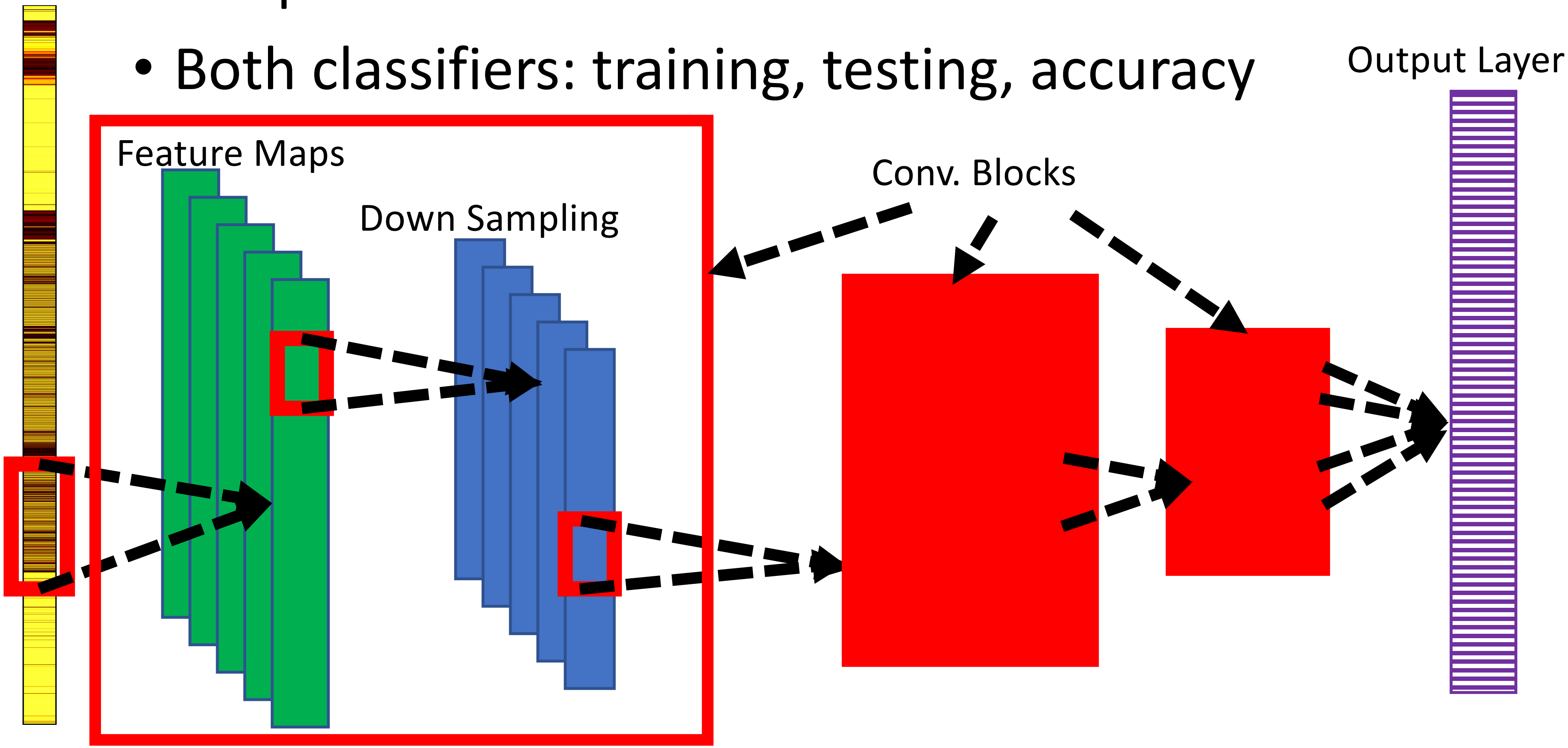
The feasibility of Website Fingerprinting (WF) attacks on Tor was assessed in a series of studies [25], [31], [19], [24], [32]. In the related works, the attack is treated as a classification problem. This problem is solved by, first, manually engineering features of traffic traces and then classifying these features with state-of-practice machine learning algorithms. Proposed approaches have been shown to achieve a classification accuracy of 91-96% correctly recognized websites [30], [24], [13] in a set of 100 websites with 100 traces per website. Their works show that finding distinctive features is essential for accurate recognition of websites. Moreover, this tasks can be costly for the adversary as he has to keep up with changes introduced in the network protocol [4], [20], [9]. The WF research community thus far has not investigated the success of an attacker who automates the feature extraction step for classification. This is the key problem that we address in this work.

An essential step of traditional machine learning is feature engineering. Feature engineering is a manual process, based on ...rt knowledge, to find a representation of raw ...s that are most relevant to the ...d to be even more ...ming

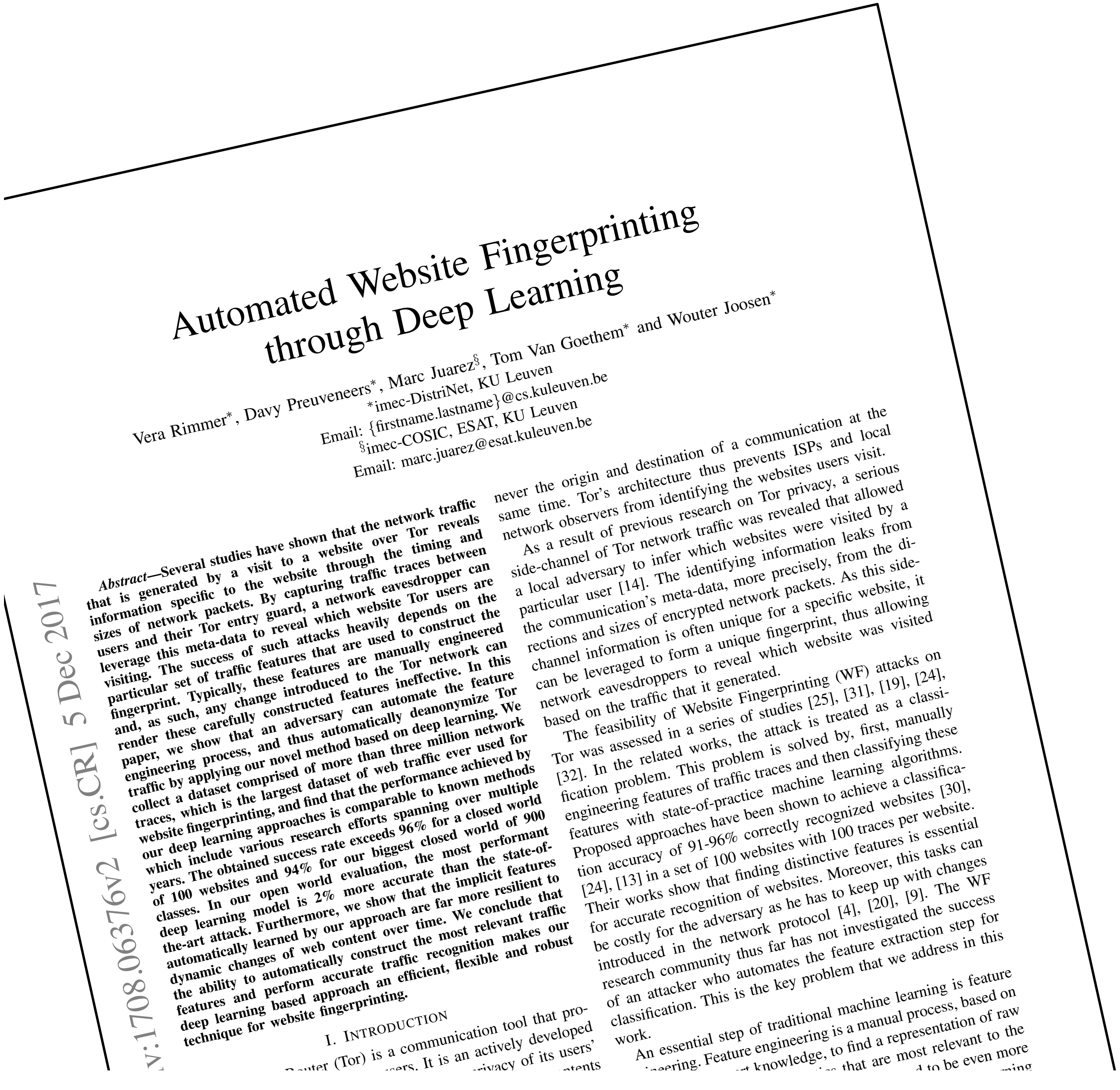
- Collect Labeled Network Traces
- Extract Features
- Train Classifier (classical/deep)
- Classify Unknown Network Traces

Classical ML vs Deep ML

- Classical ML: you choose features, classifier decides
- Deep ML: classifier chooses the features and class
- Both classifiers: training, testing, accuracy



How is WF Evaluated?



Automated Website Fingerprinting through Deep Learning

Vera Rimmer*, Davy Preuveneers*, Marc Juarez§, Tom Van Goethem* and Wouter Joosen*
*imec-DistriNet, KU Leuven
Email: {firstname.lastname}@cs.kuleuven.be
§imec-COSIC, ESAT, KU Leuven
Email: marc.juarez@esat.kuleuven.be

arXiv:1708.06376v2 [cs.CR] 5 Dec 2017

***Abstract*—Several studies have shown that the network traffic that is generated by a visit to a website over Tor reveals information specific to the website through the timing and sizes of network packets. By capturing traffic traces between users and their Tor entry guard, a network eavesdropper can leverage this meta-data to reveal which website Tor users are visiting. The success of such attacks heavily depends on the particular set of traffic features that are used to construct the fingerprint. Typically, these features are manually engineered and, as such, any change introduced to the Tor network can render these carefully constructed features ineffective. In this paper, we show that an adversary can automate the feature engineering process, and thus automatically deanonymize Tor traffic by applying our novel method based on deep learning. We collect a dataset comprised of more than three million network traces, which is the largest dataset of web traffic ever used for website fingerprinting, and find that the performance achieved by our deep learning approaches is comparable to known methods which include various research efforts spanning over multiple years. The obtained success rate exceeds 96% for a closed world of 100 websites and 94% for our biggest closed world of 900 classes. In our open world evaluation, the most performant deep learning model is 2% more accurate than the state-of-the-art attack. Furthermore, we show that the implicit features automatically learned by our approach are far more resilient to dynamic changes of web content over time. We conclude that the ability to automatically construct the most relevant traffic features and perform accurate traffic recognition makes our deep learning based approach an efficient, flexible and robust technique for website fingerprinting.**

I. INTRODUCTION

never the origin and destination of a communication at the same time. Tor's architecture thus prevents ISPs and local network observers from identifying the websites users visit.

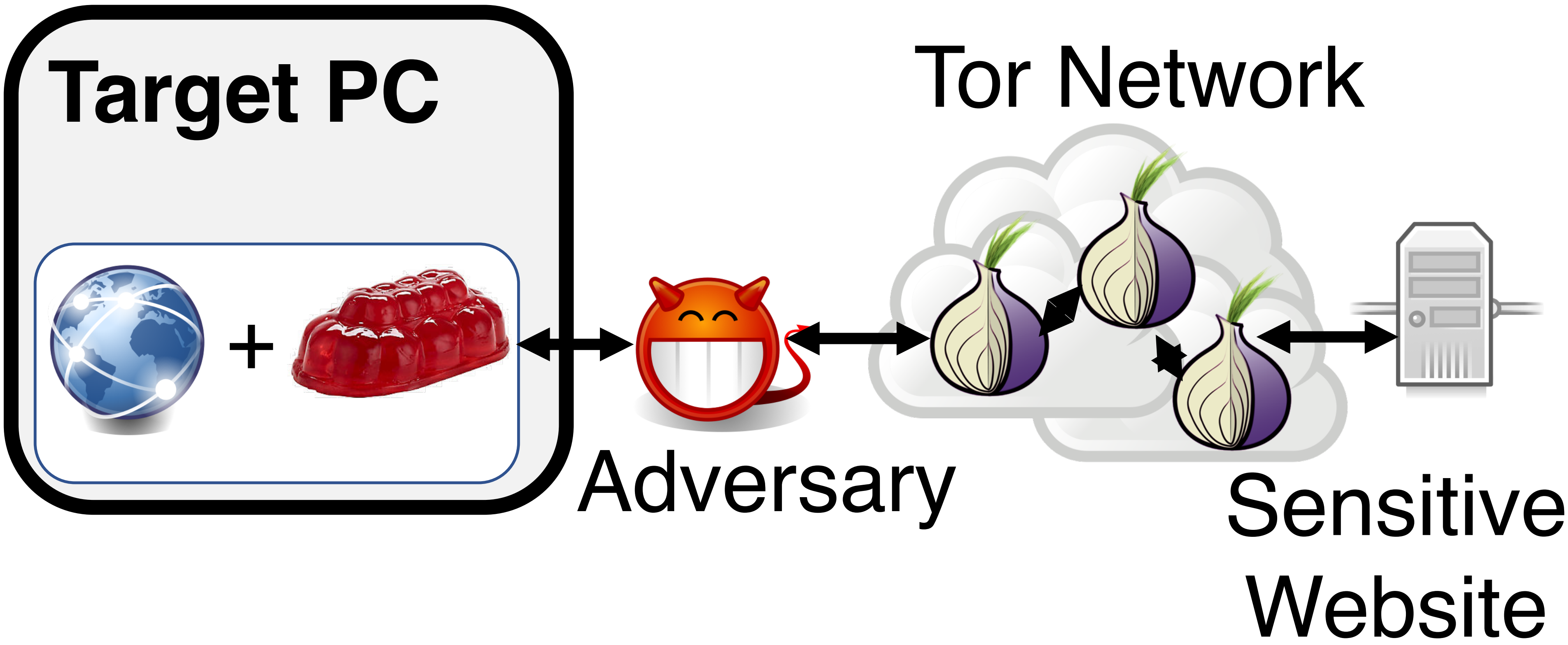
As a result of previous research on Tor privacy, a serious side-channel of Tor network traffic was revealed that allowed a local adversary to infer which websites were visited by a particular user [14]. The identifying information leaks from the communication's meta-data, more precisely, from the directions and sizes of encrypted network packets. As this side-channel information is often unique for a specific website, it can be leveraged to form a unique fingerprint, thus allowing network eavesdroppers to reveal which website was visited based on the traffic that it generated.

The feasibility of Website Fingerprinting (WF) attacks on Tor was assessed in a series of studies [25], [31], [19], [24], [32]. In the related works, the attack is treated as a classification problem. This problem is solved by, first, manually engineering features of traffic traces and then classifying these features with state-of-practice machine learning algorithms. Proposed approaches have been shown to achieve a classification accuracy of 91-96% correctly recognized websites [30], [24], [13] in a set of 100 websites with 100 traces per website. Their works show that finding distinctive features is essential for accurate recognition of websites. Moreover, this tasks can be costly for the adversary as he has to keep up with changes introduced in the network protocol [4], [20], [9]. The WF research community thus far has not investigated the success of an attacker who automates the feature extraction step for classification. This is the key problem that we address in this work.

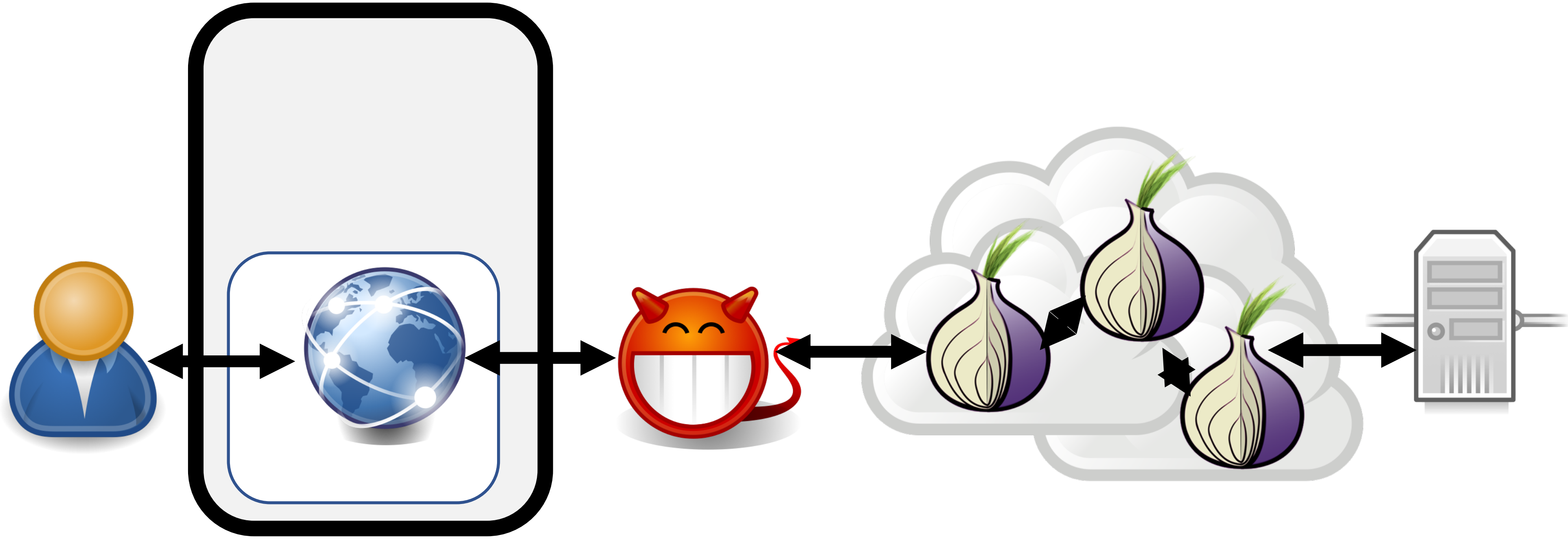
An essential step of traditional machine learning is feature engineering. Feature engineering is a manual process, based on

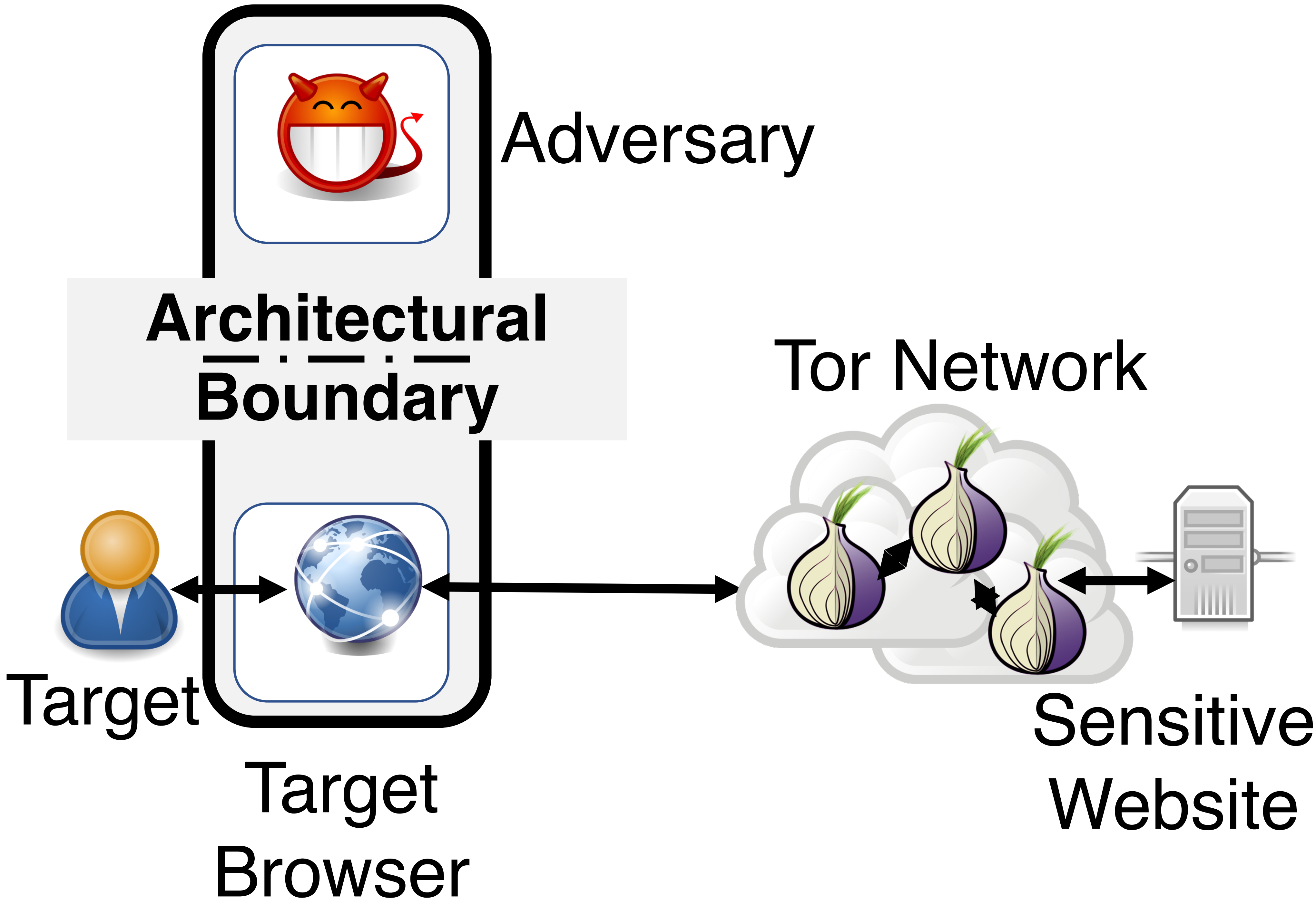
- Main metric is accuracy
- Closed World vs Open World
- Base rate is important!
- Network based WF has >90% accuracy

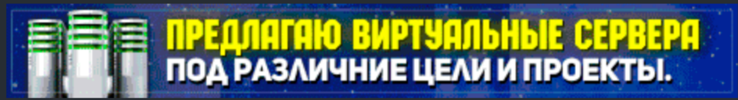
Traffic Moulding Defenses against WF



Source: lakeland.co.uk




Adversary
Architectural
Boundary
Tor Network
Target
Target
Browser
Sensitive
Website



Продажа качественных загрузок

Discussion in 'Трафик, инсталлы, загрузки - Покупка, продажа' started by sasagiant, 2 Mar 2017.

2 Mar 2017 #1

sasagiant
New Member

Joined:	22 Feb 2017
Messages:	3
Likes Received:	0
Reputations:	0

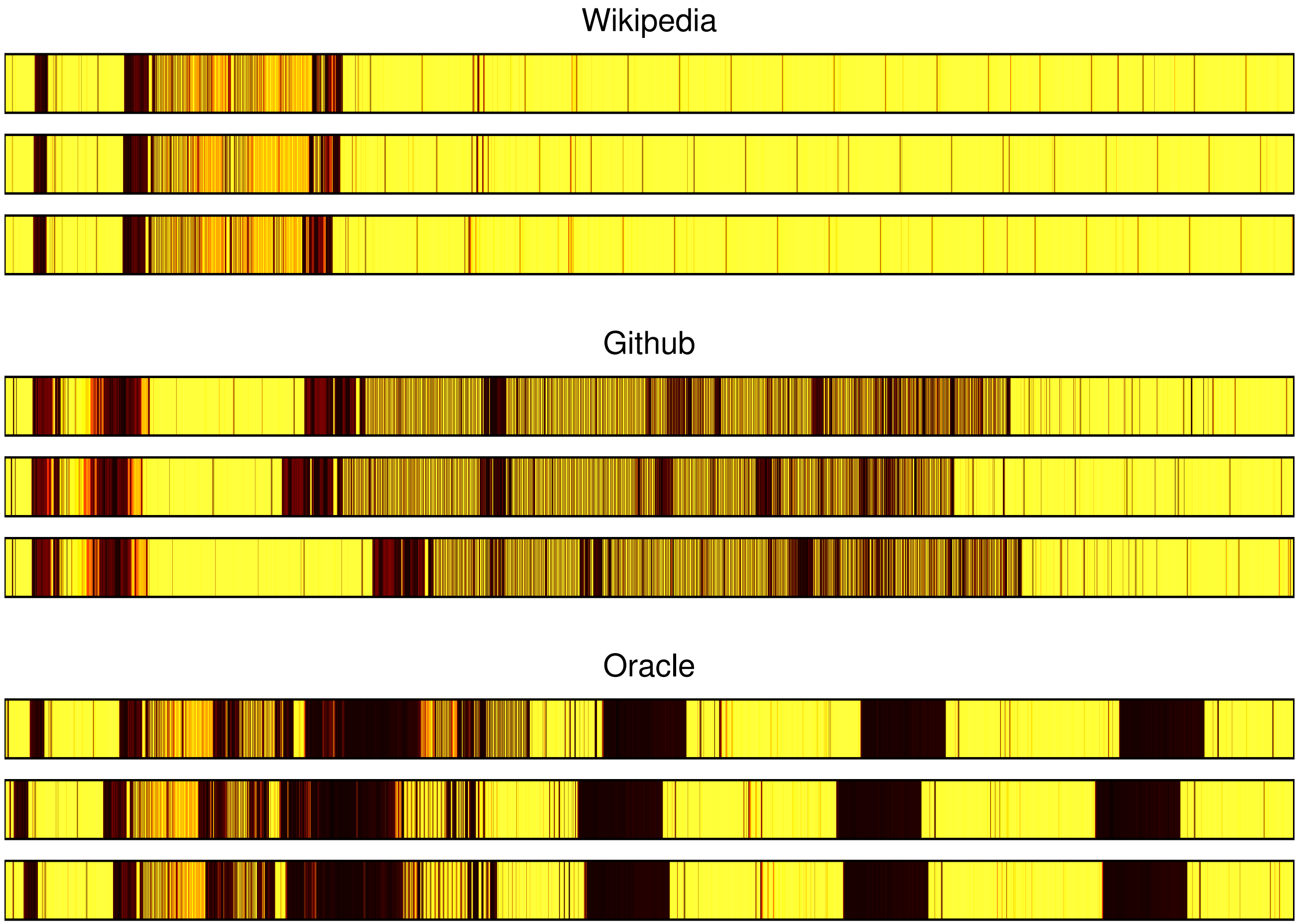
Доброго времени суток:
Представляю вашему вниманию сервис по продаже инсталлов(кроме РУ и СНГ)!!!!
Доступны большие объемы.
Просьба уточнять цены и доступное количество в личке.
Интересны оптовые закупки и прогруз на постоянной основе.
Все средства для прогруза мы предоставляем сами,exe/dll.
Старт в течении 5-15 минут.

Информация по потокам :
Тематика : миксовая
Происхождение трафа : Бирж +спам
Работаем с лоадера , мин заказ 500
доступно . микс мира Канада и юса
Цена : микс мира 100USD за 1к .(цена на обьёмы обговариваем отдельно)
Наши контакты : alen.sgor@exploit.im

тема на других форумах ..
https://fuckav.ru/showthread.php?t=32345&cdn=1

Last edited: 17 Jun 2018

Memorygrams



Cache-Based WF

Robust Website Fingerprinting Through the Cache Occupancy Channel

Anatoly Shusterman
Ben-Gurion University of the Negev
shustera@post.bgu.ac.il

Lachlan Kang
University of Adelaide
lachlan.kang@adelaide.edu.au

Yarden Haskal
Ben-Gurion Univ. of the Negev
yardenha@post.bgu.ac.il

Yosef Meltser
Ben-Gurion Univ. of the Negev
yosefmel@post.bgu.ac.il

Prateek Mittal
Princeton University
pmittal@princeton.edu

Yossi Oren
Ben-Gurion Univ. of the Negev
yos@bgu.ac.il

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

Abstract

Website fingerprinting attacks, which use statistical analysis on network traffic to compromise user privacy, have been shown to be effective even if the traffic is sent over anonymity-preserving networks such as Tor. The classical attack model used to evaluate website fingerprinting attacks assumes an *on-path adversary*, who can observe all traffic traveling between the user's computer and the secure network.

In this work we investigate these attacks under a different attack model, in which the adversary is capable of sending a small amount of malicious JavaScript code to the target user's computer. The malicious code mounts a cache side-channel attack, which exploits the effects of contention on the CPU's cache, to identify *other* websites being browsed. The effectiveness of this attack scenario has never been systematically analyzed, especially in the open-world model which assumes that the user is visiting a mix of both sensitive and non-sensitive sites.

We show that cache website fingerprinting attacks in ... feasible. Specifically, we use ma... activity reduces the effectiveness of the attack and completely eliminates it when used in the Tor Browser.

1 Introduction

Over the last decades the World Wide Web has grown from an academic exercise to a communication tool that encompasses all aspects of modern life. Users use the web to acquire information, manage their finances, conduct their social life, and more. This shift to the so called virtual life has resulted in new challenges to users' privacy. Monitoring the online behavior of users may reveal personal or sensitive information about the users, including information such as sexual orientation or political beliefs and affiliations.

Several tools have been developed to protect the online privacy of users and hide information about the websites they visit [18, 20, 71]. Prime amongst these is the Tor network [20], an overlay network of collaborating servers, called *relays*, that anonymously forward Internet traffic between users and web servers. Tor encrypts the network traffic of all of the users, and transmits it between relays in a way that prevents external observers from identifying the traffic of specific users. In addition to the network itself, the Tor ... the *Tor Browser* [82], a modified ver... that further protects ...

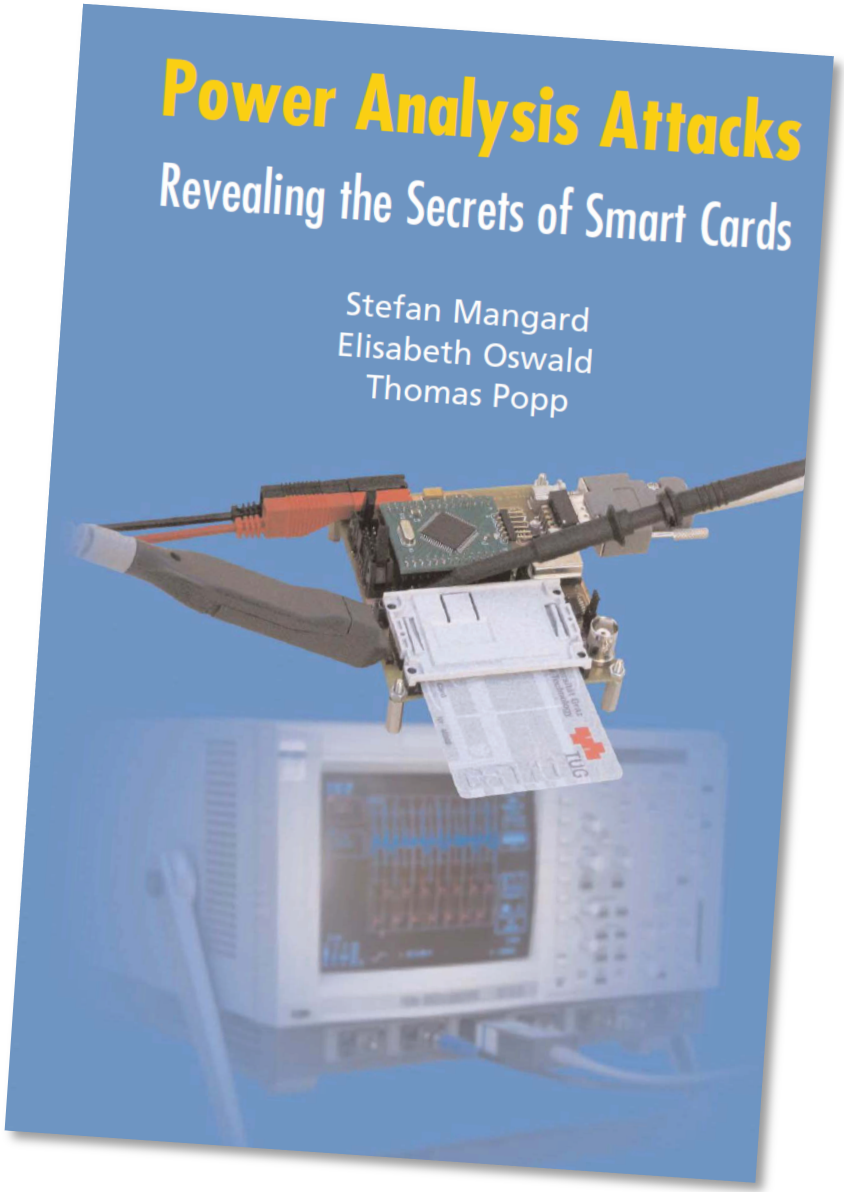
- Collect Labeled Memorygrams
- Extract Features
- Train Classifier (classical/deep)
- Classify Unknown Memorygrams
- >90% accuracy

Cache-based vs Net-based WF

Cache beats Net	Net beats Cache
Resists net countermeasures	Can be detected by victim
Robust to response caching	Depends on hardware config
Works across NICs	
Lighter attack model	

Demo

Countermeasures

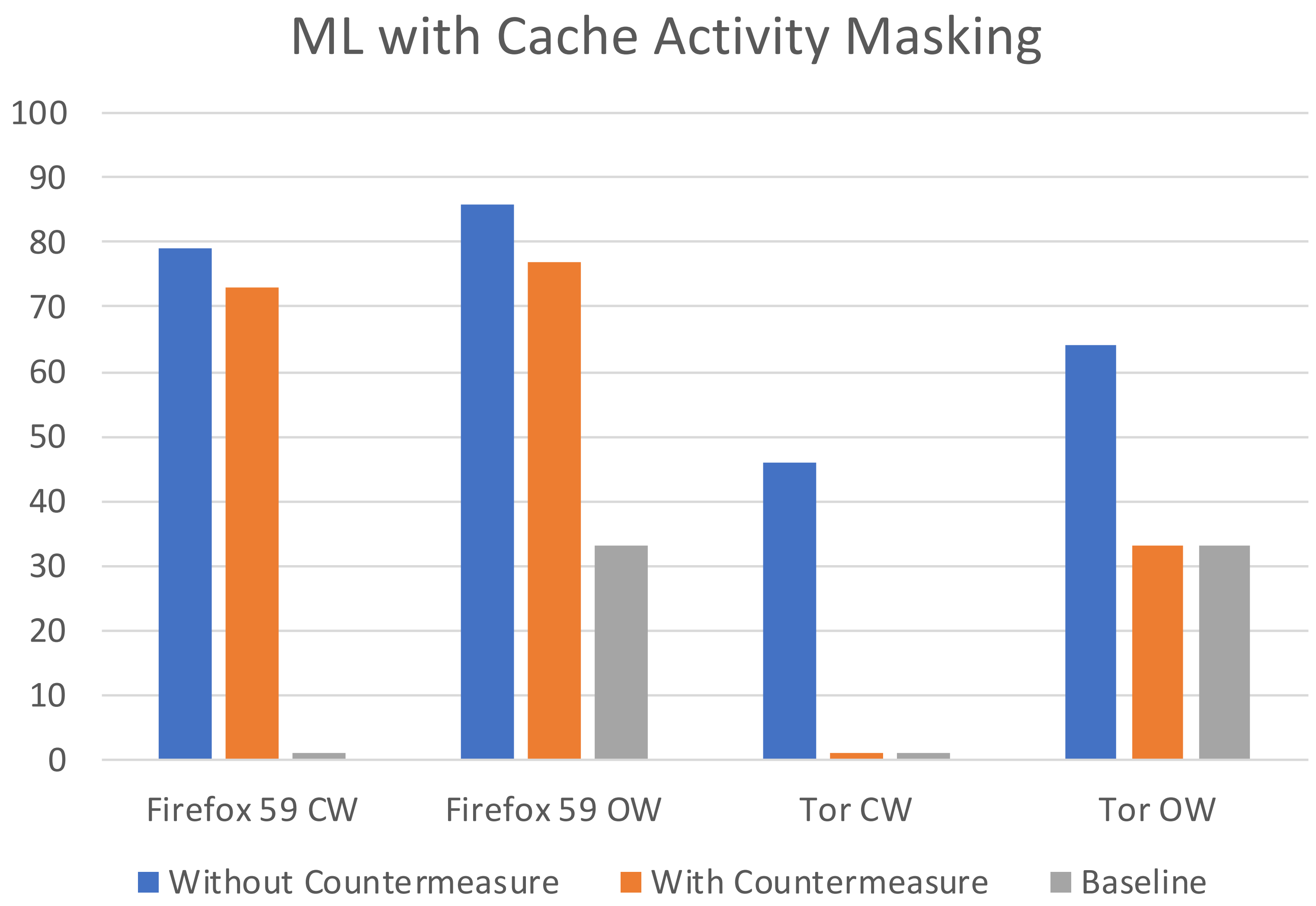


- Hiding
  - Lowering the SNR
  - Hiding in Time
  - Hiding in Amplitude
- Masking
  - Secret Invariance
  - Separation in Time
  - Separation in Space

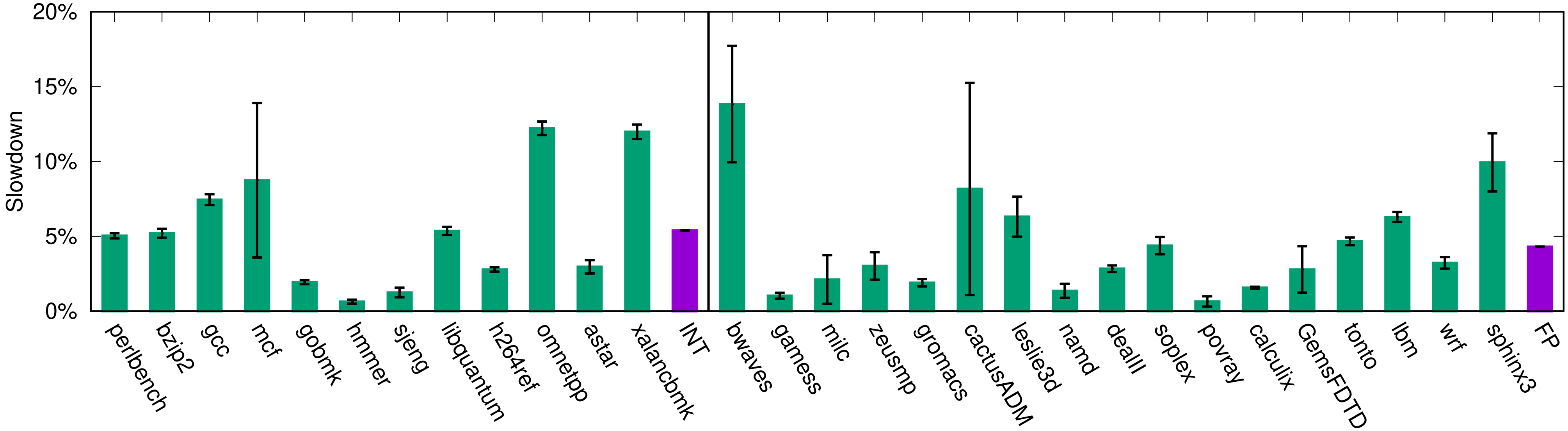
Hiding in amplitude

- Idea: run a dummy prime and probe in the background
- What is the effect on WF accuracy?
- What is the effect on performance?

Effect on Accuracy



Effect on Performance



Conclusions

- Side-channel attacks can attack **human secrets**, not just **cryptographic secrets**
- Specifically, cache-based website fingerprinting is feasible and very dangerous to user privacy
- What other secrets can we attack?
- What kind of countermeasures apply here?

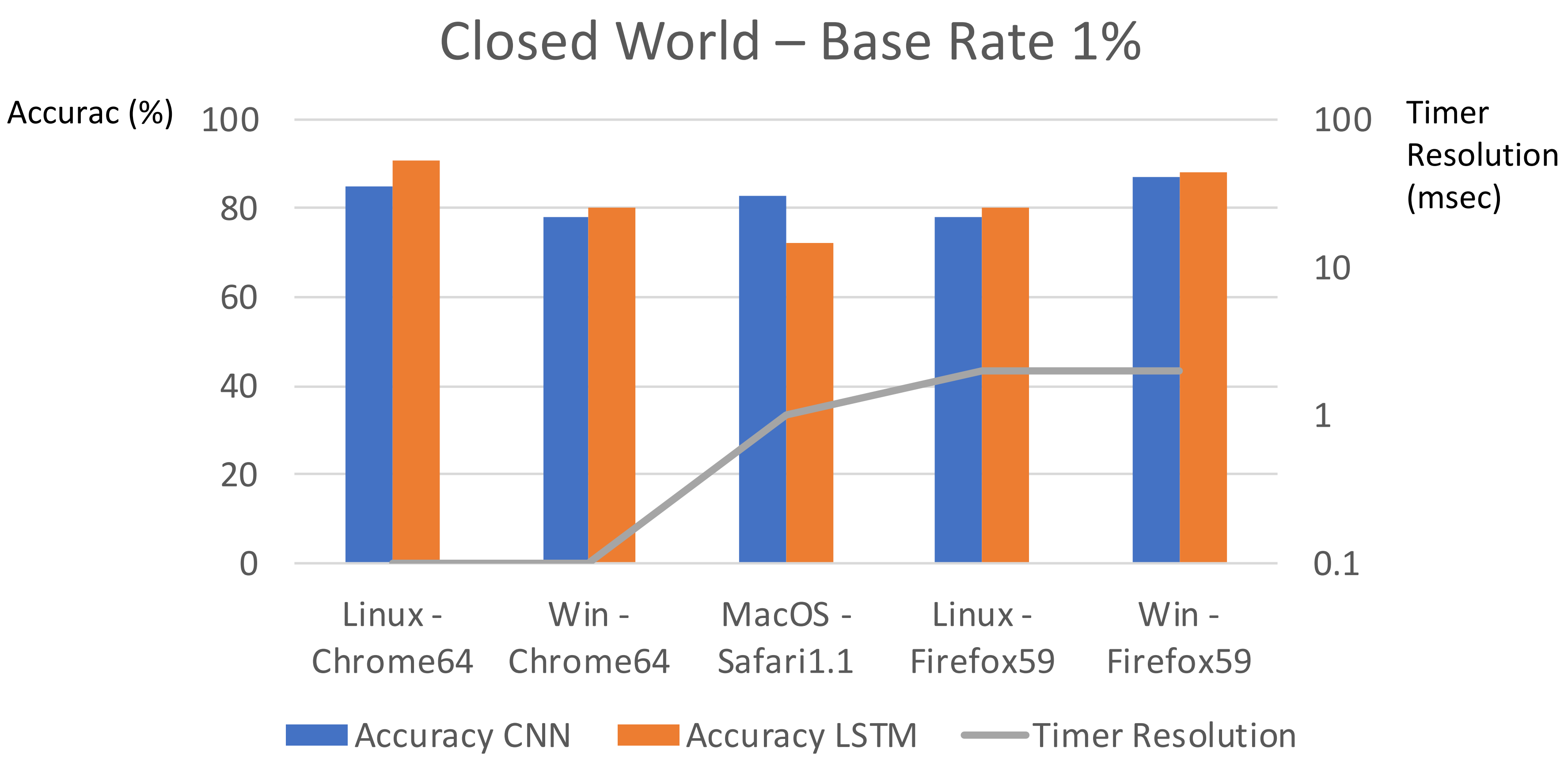
Thank you!

- Dataset freely available under CC-BY 4.0 license
- Contains:
  - Thousands of memorygrams in multiple settings
  - Associated network traces
  - Deep learning classifiers in Python

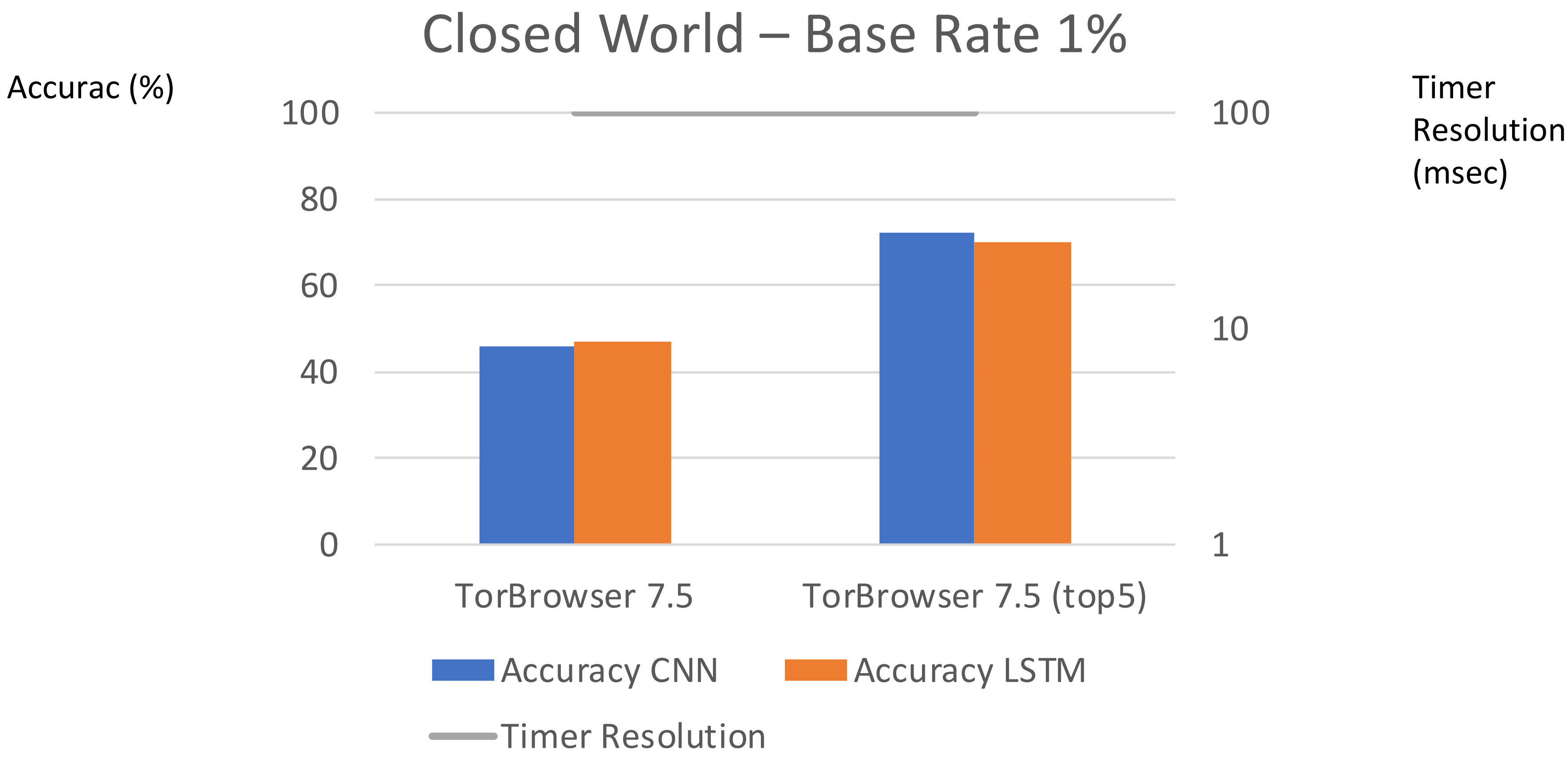
https://orenlab.sise.bgu.ac.il/publications/RobustFingerprinting

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

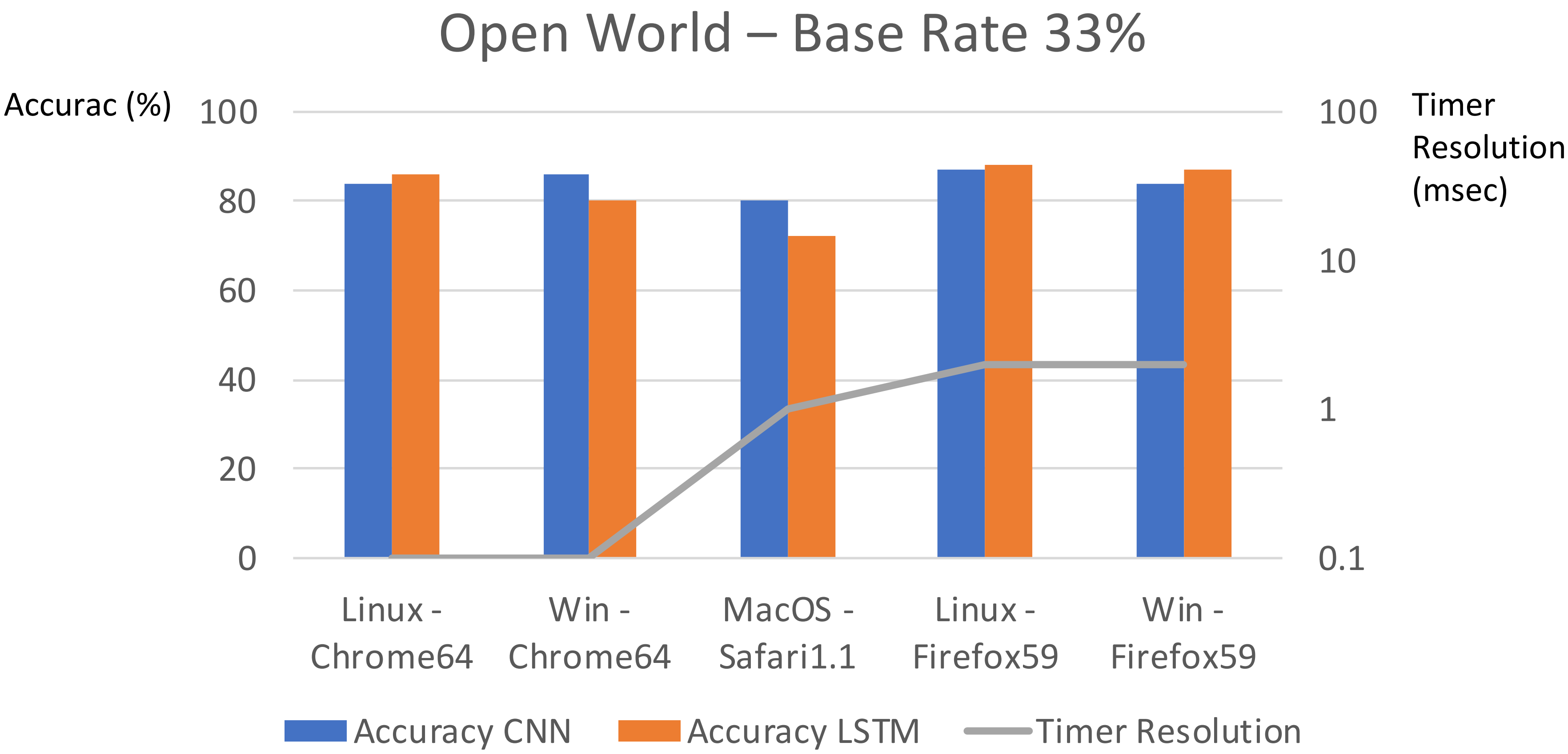
JavaScript Attack Results



JavaScript Attack Results



JavaScript Attack Results



JavaScript Attack Results