SnailLoad

Anyone on the Internet Can Learn What You're Doing

Stefan Gast, Daniel Gruss

2024-08-07

Graz University of Technology

Who are we?

Stefan Gast

PhD Student
Graz University of Technology

- @notbobbytables@infosec.exchange
- @notbobbytables
- https://stefangast.eu/

Who are we?

Stefan Gast

PhD Student
Graz University of Technology

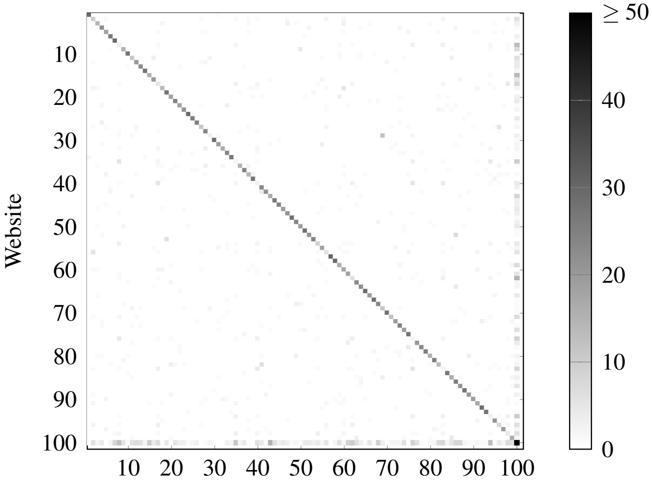
- @notbobbytables@infosec.exchange
- @notbobbytables
- https://stefangast.eu/

Daniel Gruss

Professor
Graz University of Technology

- @lavados@infosec.exchange
- @lavados
- https://gruss.cc/

We can tell which website you visit, without running anything on your system:



What are Side Channels?

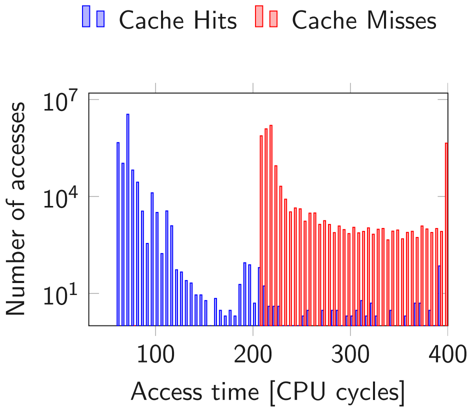
What are Side Channels?

Obtain meta-data and derive data from it

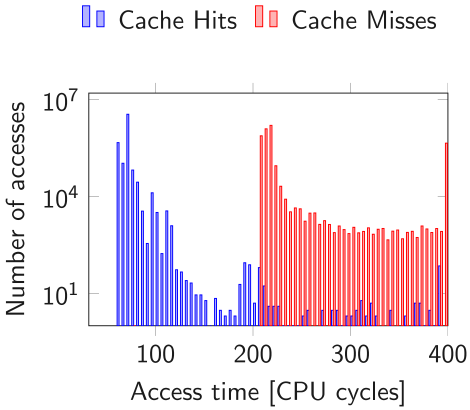
Side Channel Example

Local Timing Attack

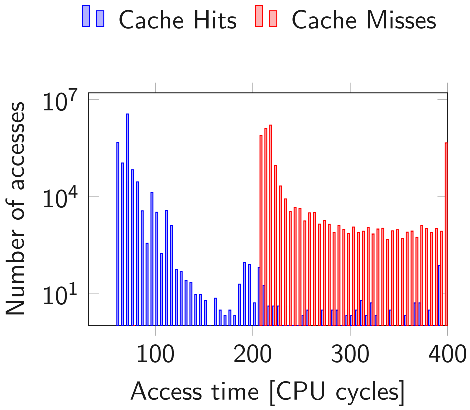
Local Timing Attack



Local Timing Attack

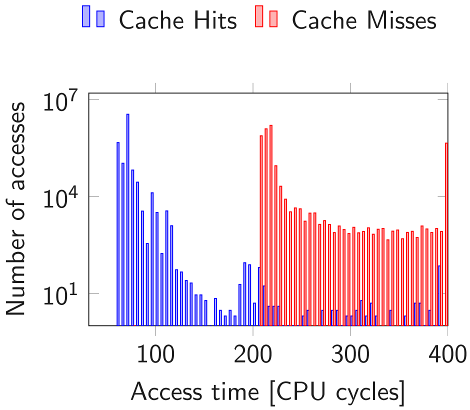


Local Timing Attack



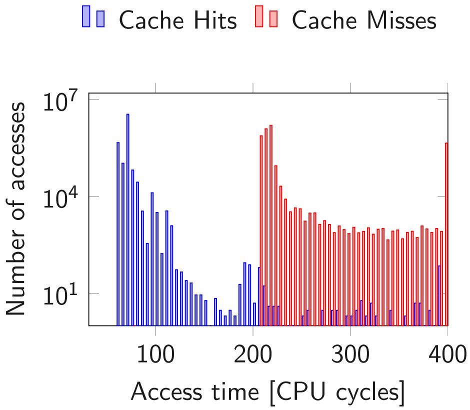
- Local → code execution

Local Timing Attack



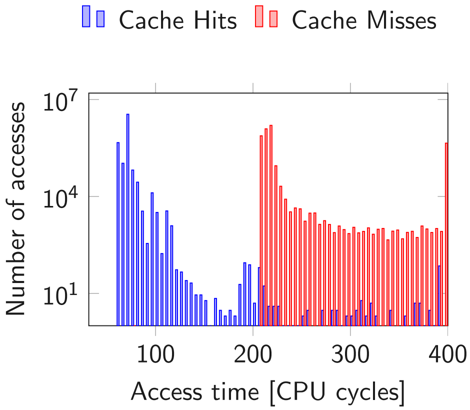
- Local → code execution
- code to use secrets

Local Timing Attack



- Local $\rightarrow$ code execution
- code to use secrets
- code to measure time

Local Timing Attack



- Local $\rightarrow$ code execution
- code to use secrets
- code to measure time
- code to exfiltrate data

Remote Timing

Remote in “remote adversary” can mean different things

Remote Timing

2018 IEEE Symposium on Security and Privacy

FPGA-Based Remote Power Side-Channel Attacks

Mark Zhao and G. Edward Suh
Computer Systems Laboratory
Cornell University
Ithaca, New York
yz424@cornell.edu, suh@ece.cornell.edu

Abstract—**The rapid adoption of heterogeneous computing has driven the integration of Field Programmable Gate Arrays (FPGAs) into cloud datacenters and flexible System-on-Chips (SoCs). This paper shows that the integrated FPGA introduces a new security vulnerability by enabling software-based power side-channel attacks without physical proximity to a target system. We first demonstrate that an on-chip power monitor can be built on a modern FPGA using ring oscillators (ROs), and characterize its ability to observe the power consumption of other modules on the FPGA or the SoC. Then, we show that the RO-based FPGA power monitor can be used for a successful power analysis attack on an RSA cryptomodule on the same FPGA. Additionally, we show that the FPGA-based**

measure the power consumption as the voltage drop across the resistor. In this paper, we demonstrate that an on-chip power monitor can be constructed using the programmable logic of an FPGA, allowing us to measure dynamic power consumption with sufficient resolution to enable power analysis attacks. In essence, the integrated FPGA opens the door for remote power analysis attacks.

This FPGA-based power side channel may be exploited in a variety of system architectures that allows an untrusted user to program a part of an FPGA. In cloud computing infrastructures, many studies from both academia and industry

Remote in “remote adversary” can mean different things

- attack from a different chip?

Remote Timing

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript

Daniel Gruss, Clémentine Maurice[†], and Stefan Mangard

Graz University of Technology, Austria

Abstract. A fundamental assumption in software security is that a memory location can only be modified by processes that may write to this memory location. However, a recent study has shown that parasitic effects in DRAM can change the content of a memory cell without accessing it, but by accessing other memory locations in a high frequency. This so-called Rowhammer bug occurs in most of today's memory modules and has fatal consequences for the security of all affected systems,

Remote in "remote adversary" can mean different things

- attack from a different chip?
- JavaScript?

Remote Timing

Hertzbleed: Turning Power Side-Channel Attacks Into Remote Timing Attacks on x86

Yingchen Wang* — *UT Austin*
Riccardo Paccagnella* — *UIUC*
Elizabeth Tang He — *UIUC*
Hovav Shacham — *UT Austin*
Christopher W. Fletcher — *UIUC*
David Kohlbrenner — *UW*

Abstract

Power side-channel attacks exploit data-dependent variations in a CPU's power consumption to leak secrets. In this paper, we show that on modern Intel (and AMD) x86 CPUs, power side-channel attacks can be *turned into* timing attacks that can be mounted without access to any power measurement interface. Our discovery is enabled by dynamic voltage and frequency scaling (DVFS). We find that, under certain circumstances, DVFS-induced variations in CPU frequency depend on the current power consumption (and hence, data) at the granularity of milliseconds. Making matters worse,

on many of today's general-purpose processors, have been abused to fingerprint websites [95], recover RSA keys [70], break KASLR [63], and even recover AES-NI keys [64].

Fortunately, software-based power-analysis attacks can be mitigated and easily detected by blocking (or restricting [10]) access to power measurement interfaces. Up until today, such a mitigation strategy would effectively reduce the attack surface to physical power analysis, a significantly smaller threat in the context of modern general-purpose x86 processors.

In this paper, we show that, on modern Intel (and AMD) x86 CPUs, power-analysis attacks can be turned into timing

Remote in "remote adversary" can mean different things

- attack from a different chip?
- JavaScript?
- network-exposed API?

Remote Timing

Off-Path TCP Hijacking in Wi-Fi Networks: A Packet-Size Side Channel Attack

Ziqiang Wang
Southeast University
ziqiangwang@seu.edu.cn

Xuewei Feng
Tsinghua University
fengxw06@126.com

Qi Li
Tsinghua University
qli01@tsinghua.edu.cn

Kun Sun
George Mason University
ksun3@gmu.edu

Yuxiang Yang
Tsinghua University
yangyx22@mails.tsinghua.edu.cn

Mengyuan Li
University of Toronto
alyssamengyuanli@gmail.com

Ganqiu Du
China software testing center
duganqiu@cstc.org.cn

Ke Xu
Tsinghua University
xuke@tsinghua.edu.cn

Jianping Wu
Tsinghua University
jianping@cernet.edu.cn

Abstract

In this paper, we unveil a fundamental side channel in Wi-Fi networks, specifically the observable frame size, which can be exploited by attackers to conduct TCP hijacking attacks. Despite the various security mechanisms (*e.g.,* WEP and WPA2/WPA3) implemented to safeguard Wi-Fi networks, our study reveals that an off path attacker can still extract suf- useful information (*e.g.,* the random sequence and acknowledgment numbers of TCP connections) from the encrypted Wi-Fi frames. Additionally, certain security policies (*e.g.,* AP isolation and rogue AP detection [33, 37]) are proposed to counteract ARP poisoning and rogue APs. Moreover, recent efforts have rectified certain implementation vulnerabilities to thwart attackers from manipulating the router's transmission

Remote in "remote adversary" can mean different things

- attack from a different chip?
- JavaScript?
- network-exposed API?
- local WiFi?

State of the Art

- local code execution $\rightarrow$ fingerprint videos

State of the Art

- local code execution → fingerprint videos
- control local gateway → precisely monitor network traffic

State of the Art

- local code execution → fingerprint videos
- control local gateway → precisely monitor network traffic
- Tor gateway → estimate network traffic

State of the Art

- local code execution → fingerprint videos
- control local gateway → precisely monitor network traffic
- Tor gateway → estimate network traffic

→ application fingerprinting

State of the Art

- local code execution → fingerprint videos
- control local gateway → precisely monitor network traffic
- Tor gateway → estimate network traffic

→ application fingerprinting

→ website fingerprinting

State of the Art

- local code execution → fingerprint videos
- control local gateway → precisely monitor network traffic
- Tor gateway → estimate network traffic

→ application fingerprinting

→ website fingerprinting

→ video fingerprinting

Internet Access Technologies

- DSL, Fiber, LTE, 5G: different throughput

Internet Access Technologies

- DSL, Fiber, LTE, 5G: different throughput
- backbone connection **has orders of magnitude higher throughput**

Stefan Gast, Daniel Gruss

- DSL, Fiber, LTE, 5G: different throughput
- backbone connection **has orders of magnitude higher throughput**

→ buffering before last mile is necessary!

Packet Buffering

Packet Buffering

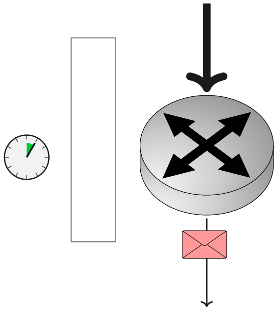


Figure 1: Connection idle

Packet Buffering

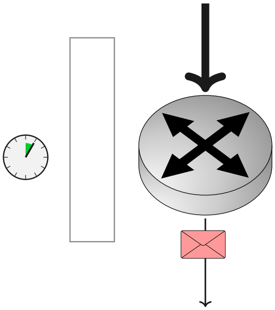


Figure 1: Connection idle

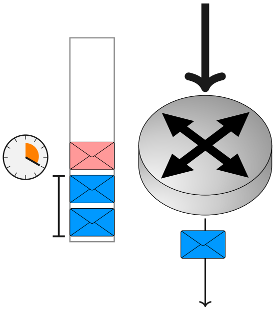


Figure 2: Connection busy

Stefan Gast, Daniel Gruss

Packet Buffering

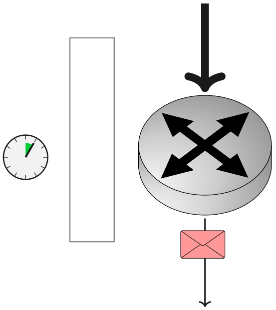


Figure 1: Connection idle

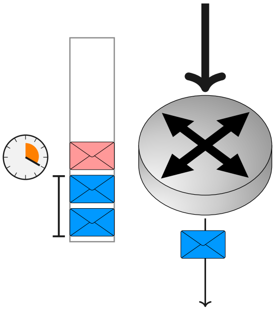


Figure 2: Connection busy

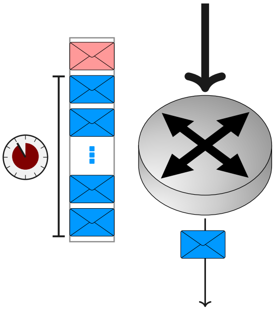


Figure 3: Bufferbloat

Network Activity Causes Latency Spikes

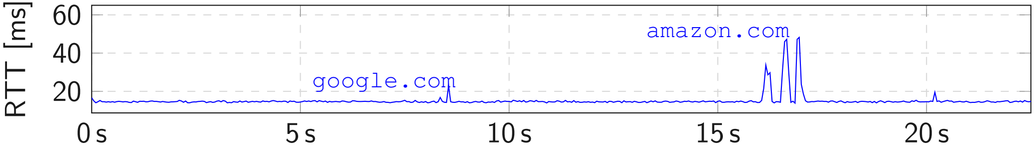


Figure 4: Same machine pinging `8.8.8.8`

Network Activity Causes Latency Spikes

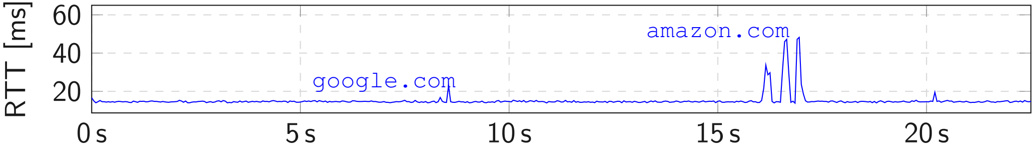


Figure 4: Same machine pinging `8.8.8.8`

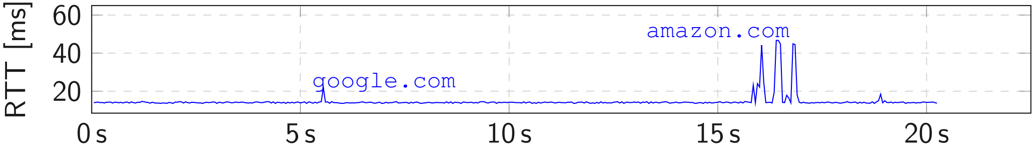


Figure 5: Different machine sharing the same internet connection pinging `8.8.8.8`

Idle and Busy Round-Trip-Times

Idle and Busy Round-Trip-Times

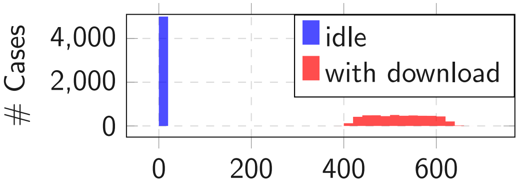


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

Idle and Busy Round-Trip-Times

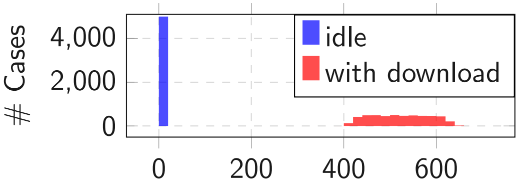


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

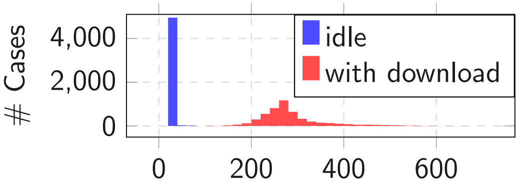


Figure 7: RTT [ms], LTE, 75 Mbit/s

Stefan Gast, Daniel Gruss

Idle and Busy Round-Trip-Times

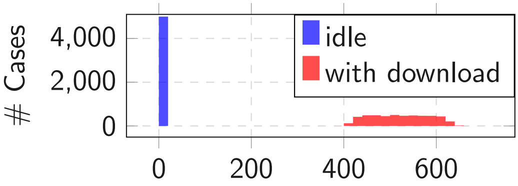


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

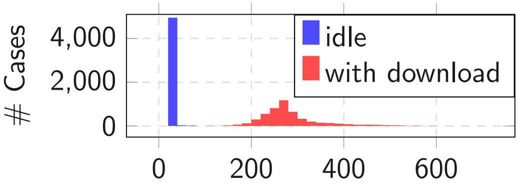


Figure 7: RTT [ms], LTE, 75 Mbit/s

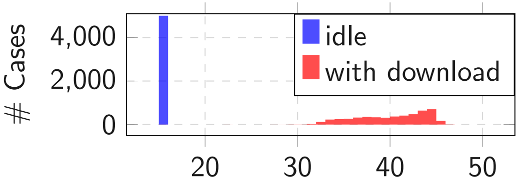


Figure 8: RTT [ms], FTTH-1, 80 Mbit/s

Idle and Busy Round-Trip-Times

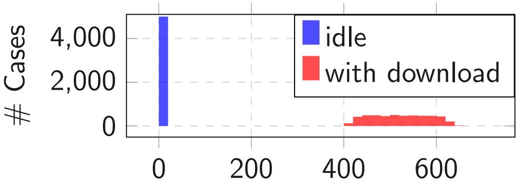


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

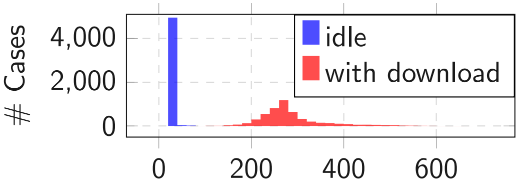


Figure 7: RTT [ms], LTE, 75 Mbit/s

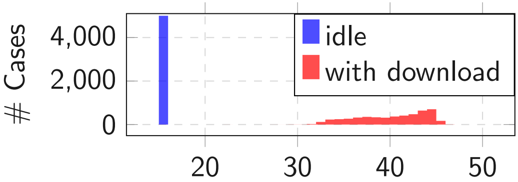


Figure 8: RTT [ms], FTTH-1, 80 Mbit/s

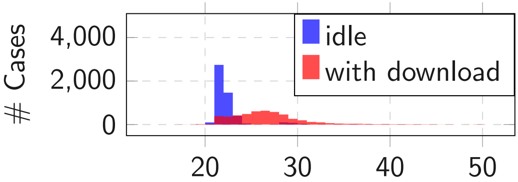
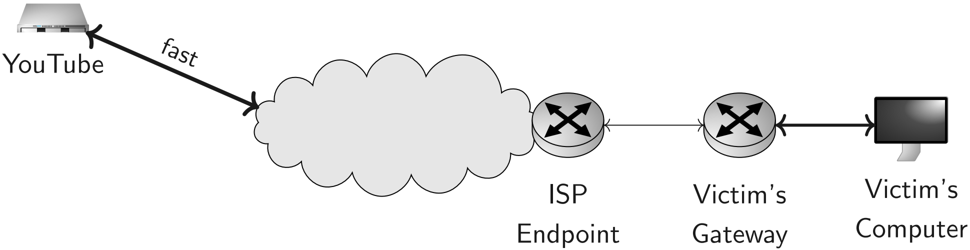
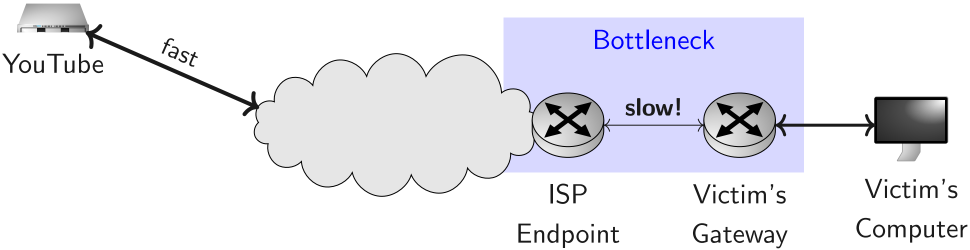


Figure 9: RTT [ms], Cable, 80 Mbit/s

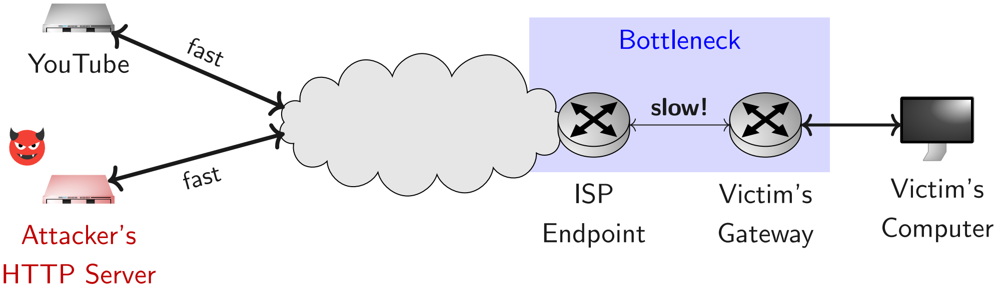
Attack Setups



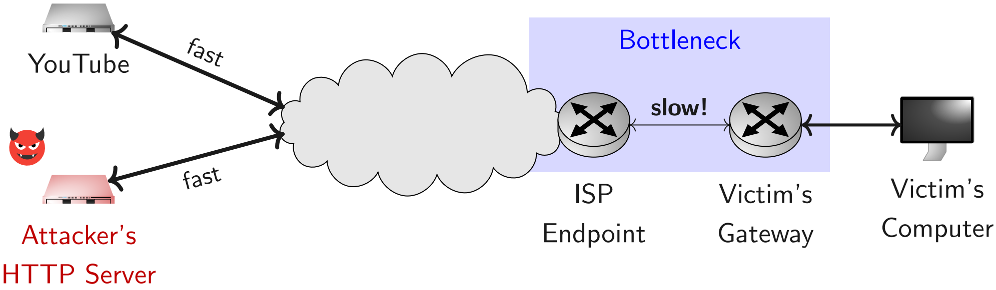
Attack Setups



Attack Setups

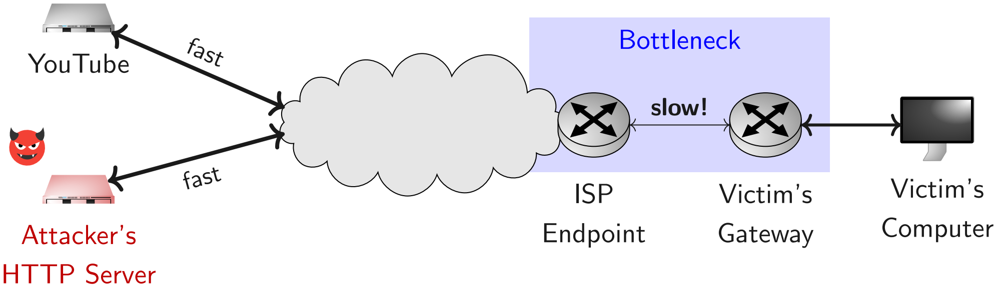


Attack Setups



- Various scenarios: Compromised websites, malicious ads, emails, and more

Attack Setups



- Various scenarios: Compromised websites, malicious ads, emails, and more
- Different ways attackers can exploit network traffic to perform attacks

Polling the Server's Send Buffer To Measure RTTs

```
begin
    acked ← false;
    start ← get_current_time();
    send(sock, b, 1, 0);
    repeat
        if ioctl(sock, SIOCOUTQ) = 0 then
            acked ← true;
        end
    until acked;
    end ← get_current_time();
    return end − start;
end
```

Polling the Server's Send Buffer To Measure RTTs

```
begin
    acked ← false;
    start ← get_current_time();
    send(sock, b, 1, 0);
    repeat
        if ioctl(sock, SIOCOUTQ) = 0 then
            acked ← true;
        end
    until acked;
    end ← get_current_time();
    return end − start;
end
```


Stefan Gast, Daniel Gruss

Polling the Server's Send Buffer To Measure RTTs

```
begin
    acked ← false;
    start ← get_current_time();
    send(sock, b, 1, 0);
    repeat
        if ioctl(sock, SIOCOUTQ) = 0 then
            acked ← true;
        end
    until acked;
    end ← get_current_time();
    return end − start;
end
```

Polling the Server's Send Buffer To Measure RTTs

```
begin
    acked ← false;
    start ← get_current_time();
    send(sock, b, 1, 0);
    repeat
        if ioctl(sock, SIOCOUTQ) = 0 then
            acked ← true;
        end
    until acked;
    end ← get_current_time();
    return end − start;
end
```

Polling the Server's Send Buffer To Measure RTTs

```
begin
    acked ← false;
    start ← get_current_time();
    send(sock, b, 1, 0);
    repeat
        if ioctl(sock, SIOCOUTQ) = 0 then
            acked ← true;
        end
    until acked;
    end ← get_current_time();
    return end − start;
end
```

Fingerprinting with Machine Learning

Fingerprinting with Machine Learning

Fingerprinting with Machine Learning

- use machine learning to analyze network traffic and infer user actions

Fingerprinting with Machine Learning

- use machine learning to analyze network traffic and infer user actions
- pre-process traces with an STFT

Fingerprinting with Machine Learning

- use machine learning to analyze network traffic and infer user actions
- pre-process traces with an STFT
- KERAS (Tensorflow)

Fingerprinting with Machine Learning

- use machine learning to analyze network traffic and infer user actions
- pre-process traces with an STFT
- KERAS (Tensorflow)
- closed-world vs. open-world

Fingerprinting with Machine Learning

- use machine learning to analyze network traffic and infer user actions
- pre-process traces with an STFT
- KERAS (Tensorflow)
- closed-world vs. open-world

Table 1: CNN Parameters

Type	Parameters	Activation
Conv2D	filters=32, kernel size=[5,5], strides=[1,1]	ReLU
MaxPooling2D	pool size=[2,2], strides=[2,2]	-
Conv2D	filters=64, kernel size=[3,3], strides=[1,1]	ReLU
MaxPooling2D	pool size=[2,2], strides=[2,2]	-
Conv2D	filters=128, kernel size=[3,3], strides=[1,1]	ReLU
MaxPooling2D	pool size=[2,2], strides=[2,2]	-
Flatten	-	-
Dense	output size=1024	ReLU
Dense	output size=512	ReLU
Dense	output size=10	Softmax

Video Fingerprinting

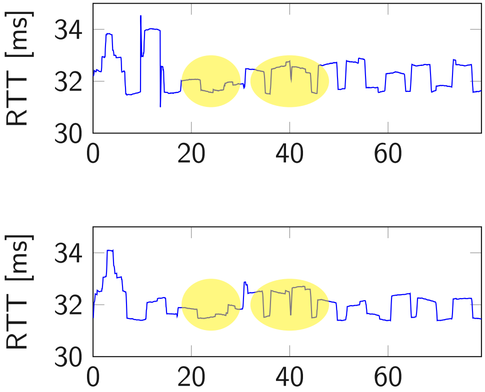


Figure 10: Video A, Time in seconds on x axis

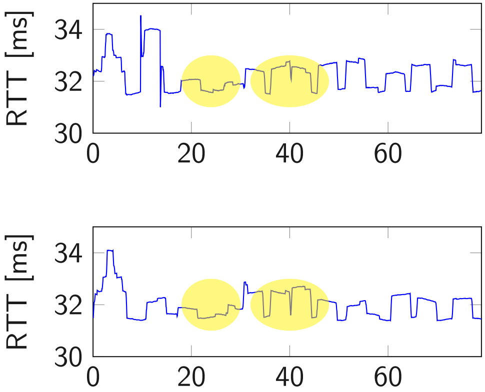


Figure 10: Video A, Time in seconds on x axis

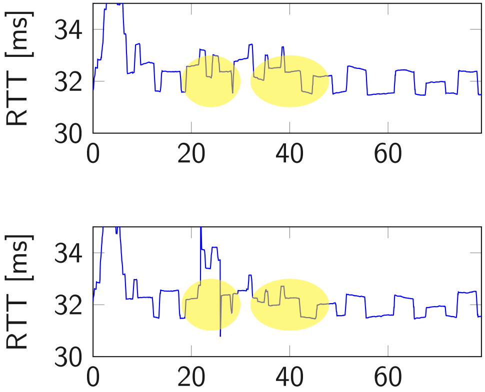
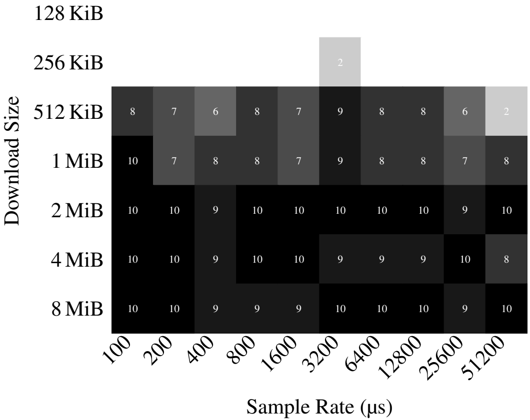
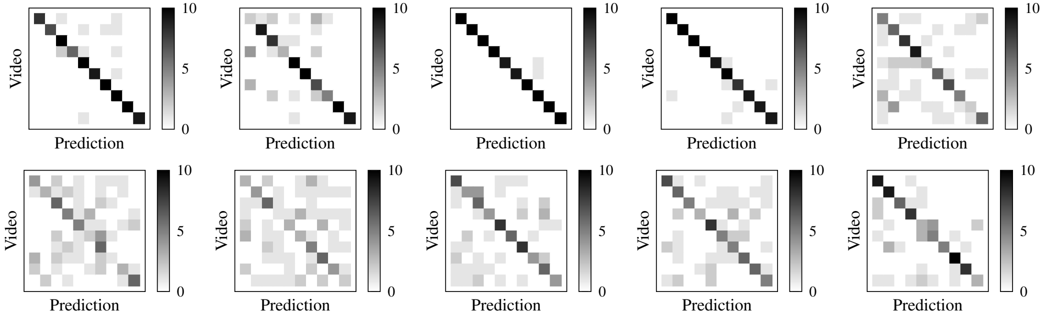


Figure 11: Video B, Time in seconds on x axis

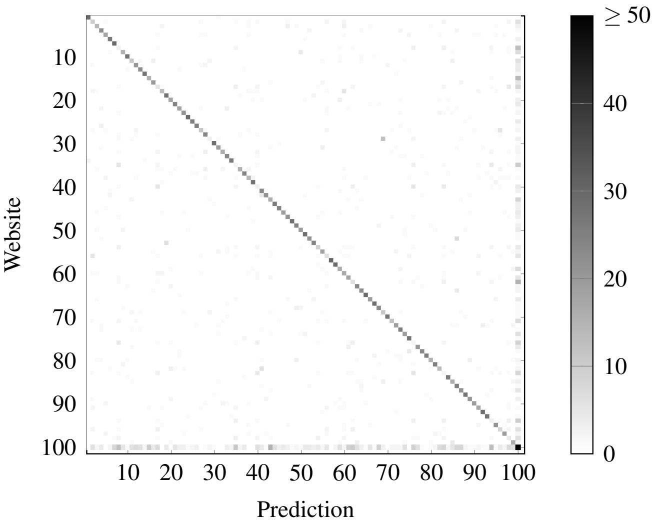
How large does the website have to be?



Video Fingerprinting on 10 different connections

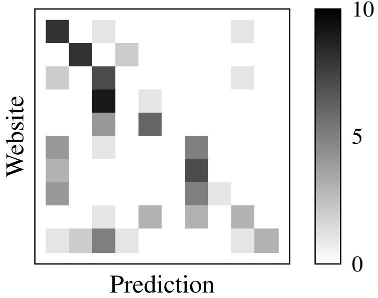


Top-100 Open-World Website Fingerprinting

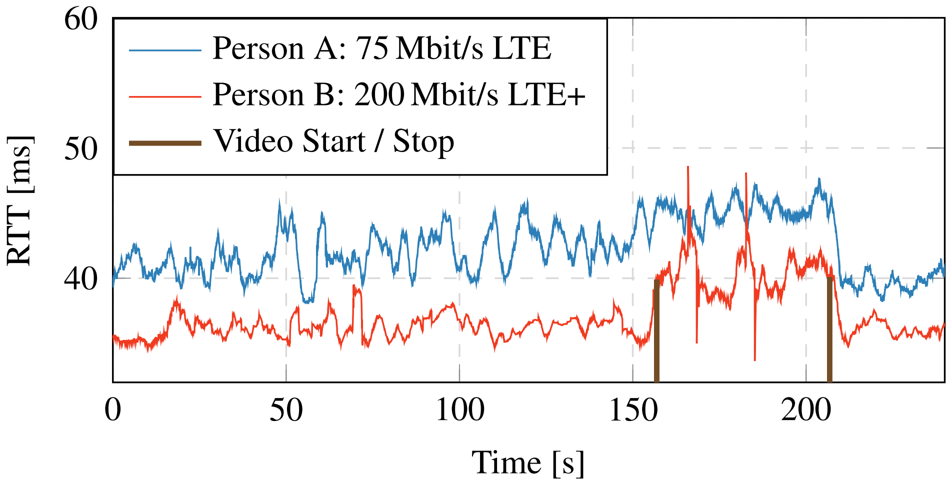


Stefan Gast, Daniel Gruss

Cross-Connection Website Fingerprinting

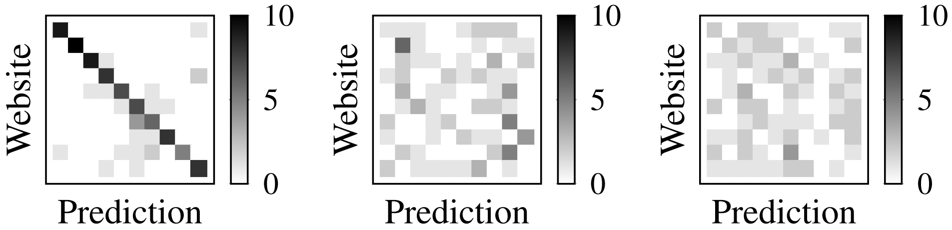


Live Demo



60
50
40
RTT [ms]
0
50
100
150
200
Time [s]
Person A: 75 Mbit/s LTE
Person B: 200 Mbit/s LTE+
Video Start / Stop

Impact of Noise on Website Fingerprinting



Context, Impact, Disclosure

- SnailLoad is a generic problem of heterogenous networks (with different throughputs)

Context, Impact, Disclosure

- SnailLoad is a generic problem of heterogenous networks (with different throughputs)
- Many "remote" attacks can now be transformed to truly remote attacks

Context, Impact, Disclosure

- SnailLoad is a generic problem of heterogenous networks (with different throughputs)
- Many “remote” attacks can now be transformed to truly remote attacks
- We disclosed to Google / YouTube

Context, Impact, Disclosure

- SnailLoad is a generic problem of heterogenous networks (with different throughputs)
- Many "remote" attacks can now be transformed to truly remote attacks
- We disclosed to Google / YouTube
  - they investigated the issue for several weeks

Context, Impact, Disclosure

- SnailLoad is a generic problem of heterogenous networks (with different throughputs)
- Many “remote” attacks can now be transformed to truly remote attacks
- We disclosed to Google / YouTube
  - they investigated the issue for several weeks
  - concluded that it is a generic problem

Take Aways (Black Hat Sound Bytes)

- Any connection to a remote server can obtain high-resolution traces of your activity

Take Aways (Black Hat Sound Bytes)

- Any connection to a remote server can obtain high-resolution traces of your activity
- Traces can leak websites and videos watched

Take Aways (Black Hat Sound Bytes)

- Any connection to a remote server can obtain high-resolution traces of your activity
- Traces can leak websites and videos watched
- Throughput difference is the root cause $\rightarrow$ not trivial to fix

Acknowledgments

This research was made possible by generous funding from:

Funded by the European Union

fssec erc European Research Council Established by the European Commission

SPyCoDe FWF Der Wissenschaftsfonds.

NeRAM FWF Der Wissenschaftsfonds. DFG Deutsche Forschungsgemeinschaft

Red Hat

Google

intel

Supported in part by the European Research Council (ERC project FSSec 101076409) and the Austrian Science Fund (FWF SFB project SPyCoDe 10.55776/F85 and FWF project NeRAM I6054). Additional funding was provided by generous gifts from Red Hat, Google, and Intel. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding parties.

SnailLoad

Anyone on the Internet Can Learn What You're Doing

Stefan Gast, Daniel Gruss

2024-08-07

Graz University of Technology