Side-Channel Security

Chapter 7: Network Side Channels

Stefan Gast

2025-04-03

Introduction

What to Attack?

Shared Resources

- Memory
- CPU

What to Attack?

Shared Resources

- Memory
- CPU
- Internet Connection

Demo: Network Traffic Depends on Activity

Website Fingerprinting

Every website causes a characteristic traffic pattern – a fingerprint:

Website Fingerprinting

Every website causes a characteristic traffic pattern – a fingerprint:

- Hintz, 2003 [Hin03]: asset transfer sizes

Website Fingerprinting

Every website causes a characteristic traffic pattern – a fingerprint:

- Hintz, 2003 [Hin03]: asset transfer sizes
- Panchenko et al. , 2011 [Pan+11]: packet sizes, directions, order

Website Fingerprinting

Every website causes a characteristic traffic pattern – a fingerprint:

- Hintz, 2003 [Hin03]: asset transfer sizes
- Panchenko et al. , 2011 [Pan+11]: packet sizes, directions, order
- Rimmer et al. , 2017 [Rim+17]: traffic shape (packet sizes, directions, timings), CNN classifier

Website Fingerprinting

Every website causes a characteristic traffic pattern – a fingerprint:

- Hintz, 2003 [Hin03]: asset transfer sizes
- Panchenko et al. , 2011 [Pan+11]: packet sizes, directions, order
- Rimmer et al. , 2017 [Rim+17]: traffic shape (packet sizes, directions, timings), CNN classifier
- ...

Website Fingerprinting

Every website causes a characteristic traffic pattern – a fingerprint:

- Hintz, 2003 [Hin03]: asset transfer sizes
- Panchenko et al. , 2011 [Pan+11]: packet sizes, directions, order
- Rimmer et al. , 2017 [Rim+17]: traffic shape (packet sizes, directions, timings), CNN classifier
- ...

→ attacker-in-the-middle, mostly used against privacy-enhancing tunnels

Video Fingerprinting

Which video segment uses more bandwidth?

https://www.youtube.com/watch?v=LNI8rnxxVvQ

Video Fingerprinting

- Dynamic Adaptive Streaming over HTTP (DASH) [ISO22]

Video Fingerprinting

- Dynamic Adaptive Streaming over HTTP (DASH) [ISO22]
- usually encrypted

Video Fingerprinting

- Dynamic Adaptive Streaming over HTTP (DASH) [ISO22]
- usually encrypted
- split video into segments with a few seconds duration

Video Fingerprinting

- Dynamic Adaptive Streaming over HTTP (DASH) [ISO22]
- usually encrypted
- split video into segments with a few seconds duration
- send segments on demand

Video Fingerprinting

- Dynamic Adaptive Streaming over HTTP (DASH) [ISO22]
- usually encrypted
- split video into segments with a few seconds duration
- send segments on demand
- segment durations and sizes depend on content

Video Fingerprinting

- Dynamic Adaptive Streaming over HTTP (DASH) [ISO22]
- usually encrypted
- split video into segments with a few seconds duration
- send segments on demand
- segment durations and sizes depend on content

→ fingerprint!

Video Fingerprinting

- Reed and Kranch, 2017 [RK17]: Netflix
- Schuster et al. , 2017 [SST17]: YouTube, Netflix, Amazon, Vimeo
- Gu et al. , 2018 [Gu+18]: self-hosted DASH server
- ...

Video Fingerprinting

- Reed and Kranch, 2017 [RK17]: Netflix
- Schuster et al. , 2017 [SST17]: YouTube, Netflix, Amazon, Vimeo
- Gu et al. , 2018 [Gu+18]: self-hosted DASH server
- …

→ attacker-in-the-middle or with JavaScript

Other Traffic Analysis Attacks

- SSH keystroke timings [SWT01]

Other Traffic Analysis Attacks

- SSH keystroke timings [SWT01]
- deanonymization of Tor users [RSG98; AYR15; Wan+11]

Other Traffic Analysis Attacks

- SSH keystroke timings [SWT01]
- deanonymization of Tor users [RSG98; AYR15; Wan+11]
- language [Wri+07] and phonemes [Whi+11] of VoIP calls

Other Traffic Analysis Attacks

- SSH keystroke timings [SWT01]
- deanonymization of Tor users [RSG98; AYR15; Wan+11]
- language [Wri+07] and phonemes [Whi+11] of VoIP calls
- other privacy-critical information [Che+10; LM18]

SnailLoad: Remote Traffic Analysis via TCP [Gas+24]

Some of you probably know the effect…

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**

Internet Access Technologies

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

→ buffering before last mile is necessary!

Packet Buffering

Packet Buffering

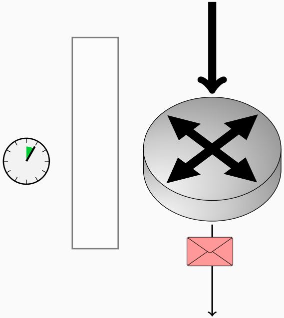


Figure 1: Connection idle

Packet Buffering

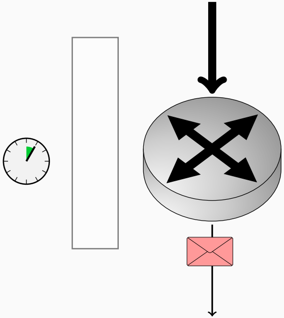


Figure 1: Connection idle

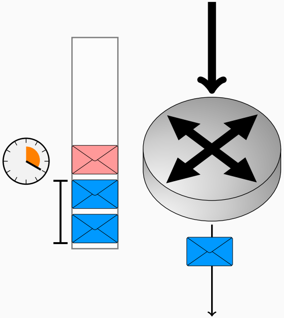


Figure 2: Connection busy

Packet Buffering

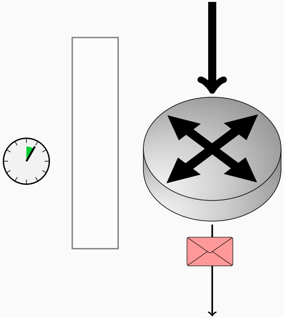


Figure 1: Connection idle

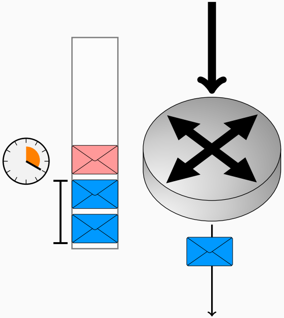


Figure 2: Connection busy

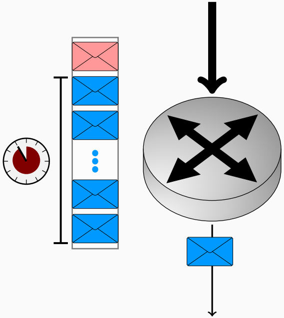


Figure 3: Bufferbloat

Network Activity Causes Latency Spikes

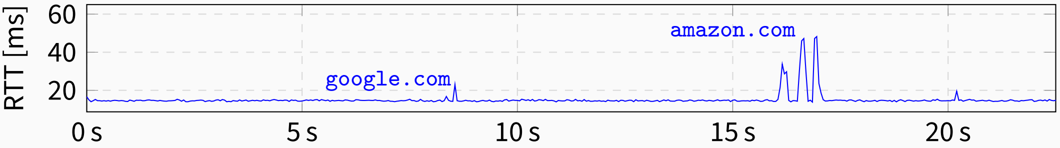


Figure 4: Same machine pinging `8.8.8.8`

Network Activity Causes Latency Spikes

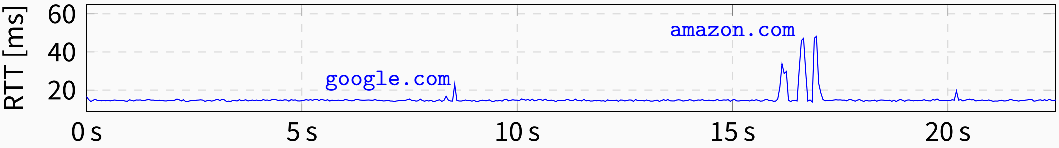


Figure 4: Same machine pinging `8.8.8.8`

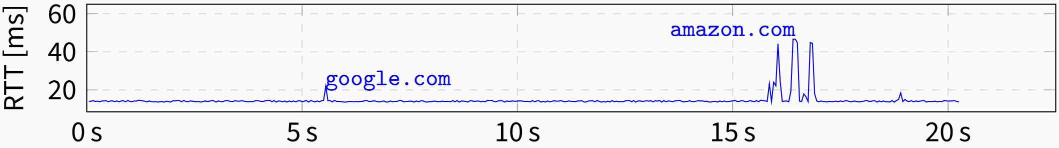


Figure 5: Different machine sharing the same internet connection pinging `8.8.8.8`

Idle and Busy Round-Trip-Times

Idle and Busy Round-Trip-Times

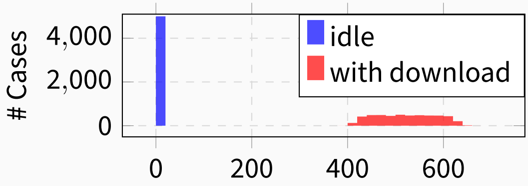


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

Idle and Busy Round-Trip-Times

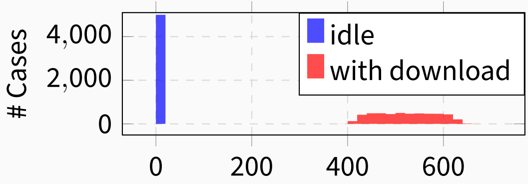


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

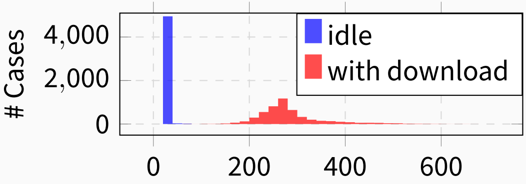


Figure 7: RTT [ms], LTE, 75 Mbit/s

Idle and Busy Round-Trip-Times

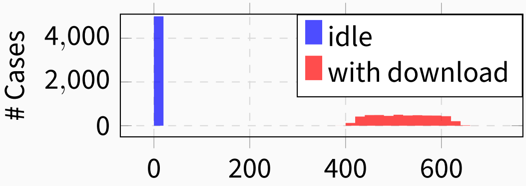


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

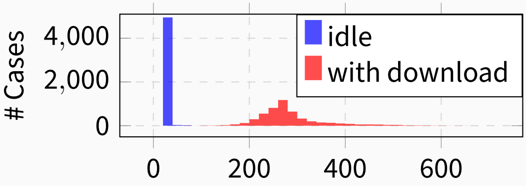


Figure 7: RTT [ms], LTE, 75 Mbit/s

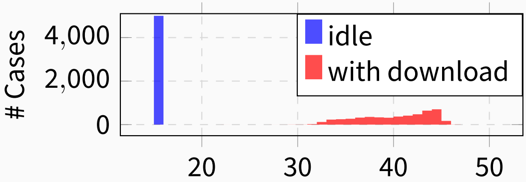


Figure 8: RTT [ms], FTTH-1, 80 Mbit/s

Idle and Busy Round-Trip-Times

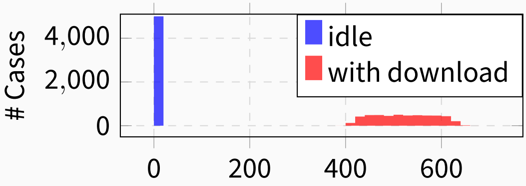


Figure 6: RTT [ms], ADSL-1, 50 Mbit/s

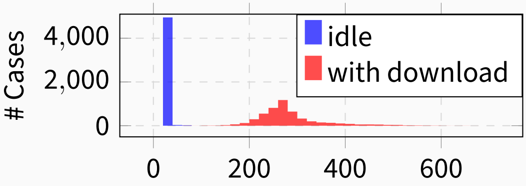


Figure 7: RTT [ms], LTE, 75 Mbit/s

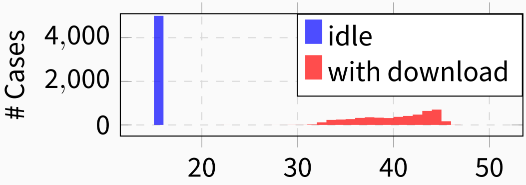


Figure 8: RTT [ms], FTTH-1, 80 Mbit/s

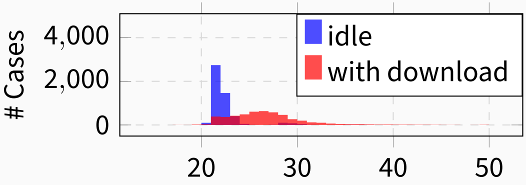
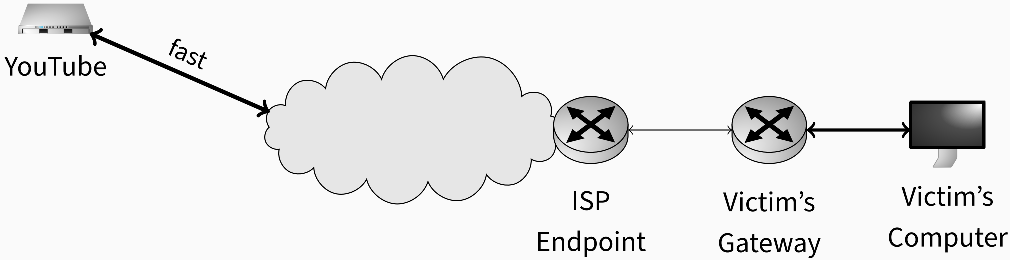
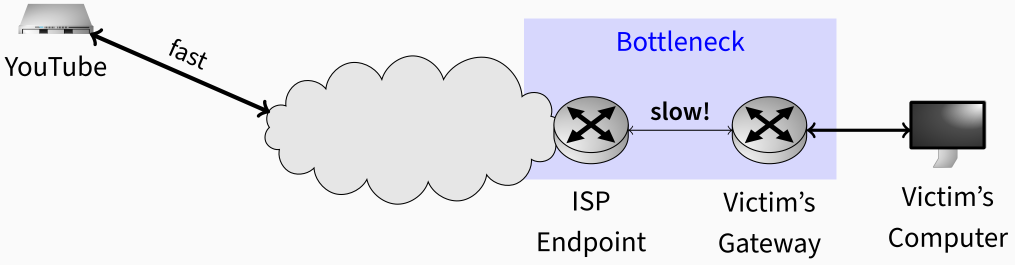


Figure 9: RTT [ms], Cable, 80 Mbit/s

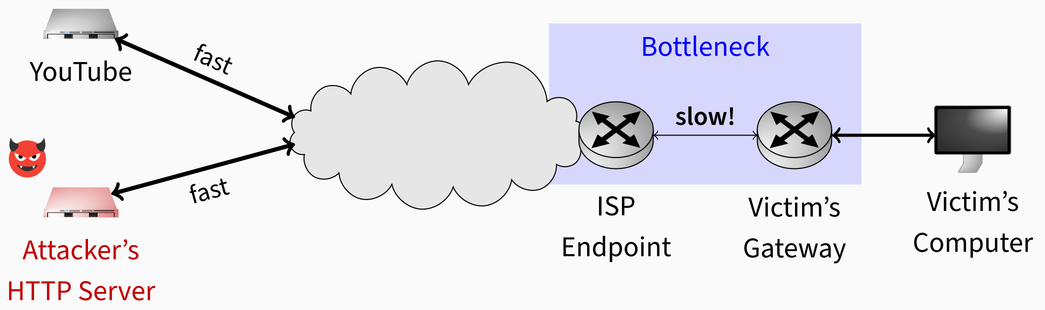
Attack Setup



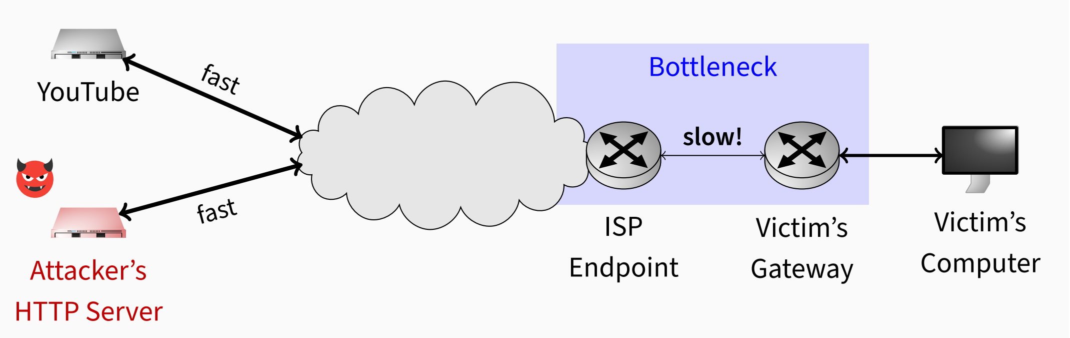
Attack Setup



Attack Setup

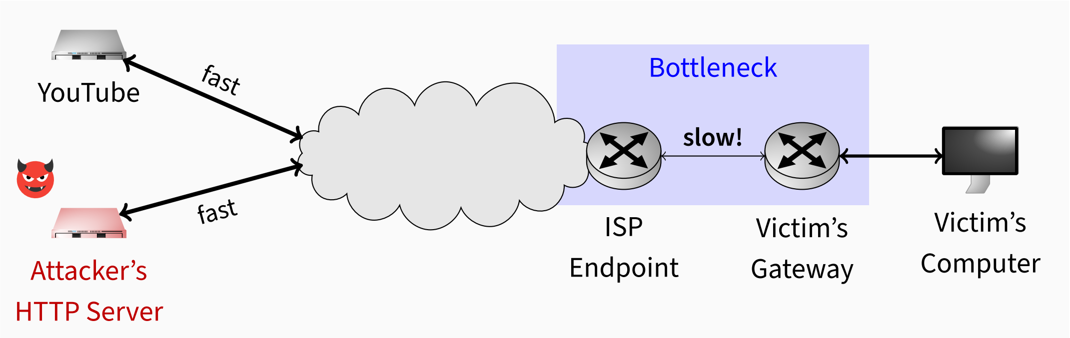


Attack Setup



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

Attack Setup



- 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
```

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

- 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)

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

Video Fingerprinting

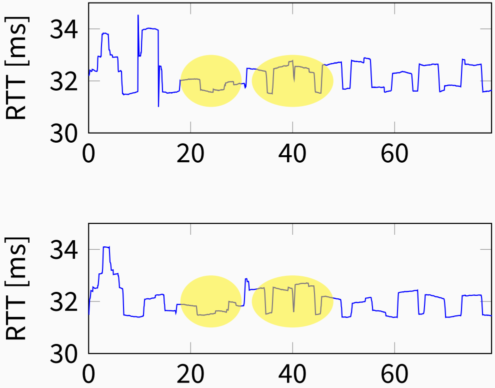


Figure 10: Video A, Time in seconds on x axis

Video Fingerprinting

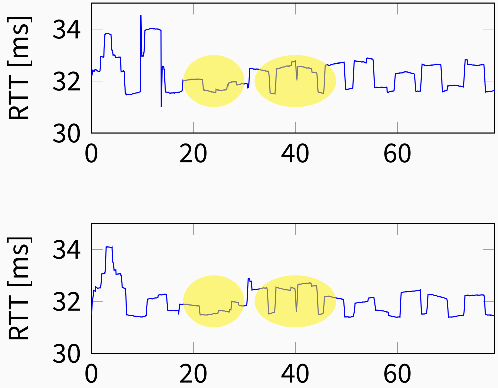


Figure 10: Video A, Time in seconds on x axis

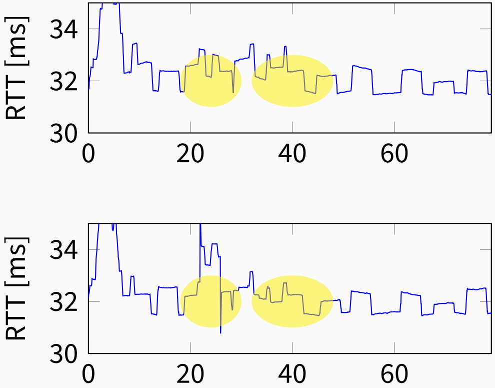
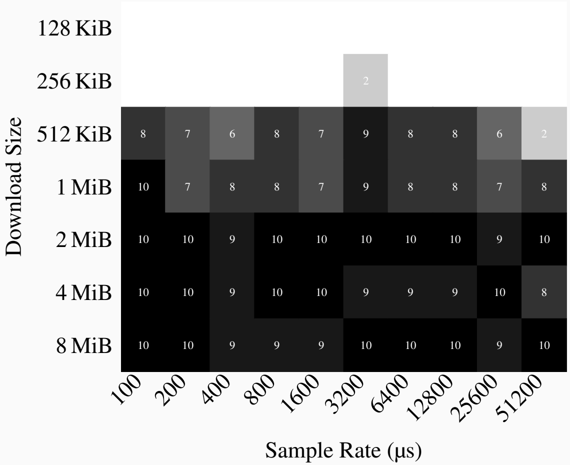
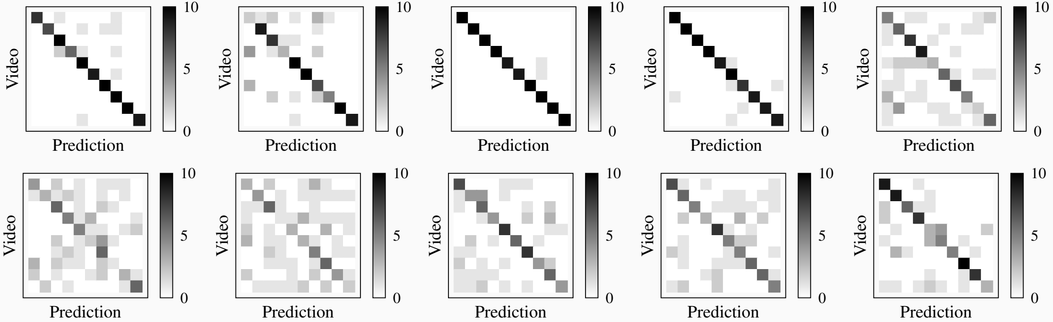


Figure 11: Video B, Time in seconds on x axis

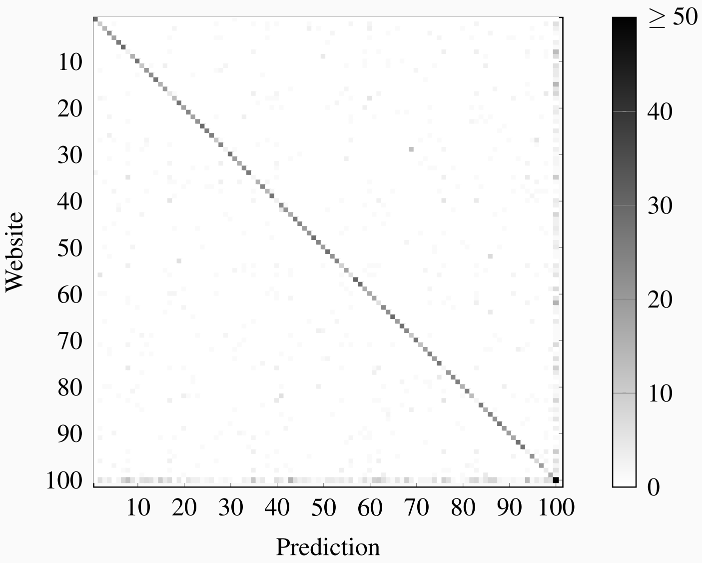
How large does the website have to be?



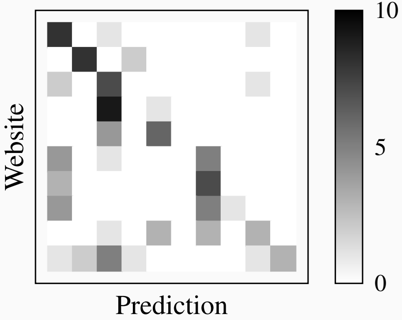
Video Fingerprinting on 10 different connections



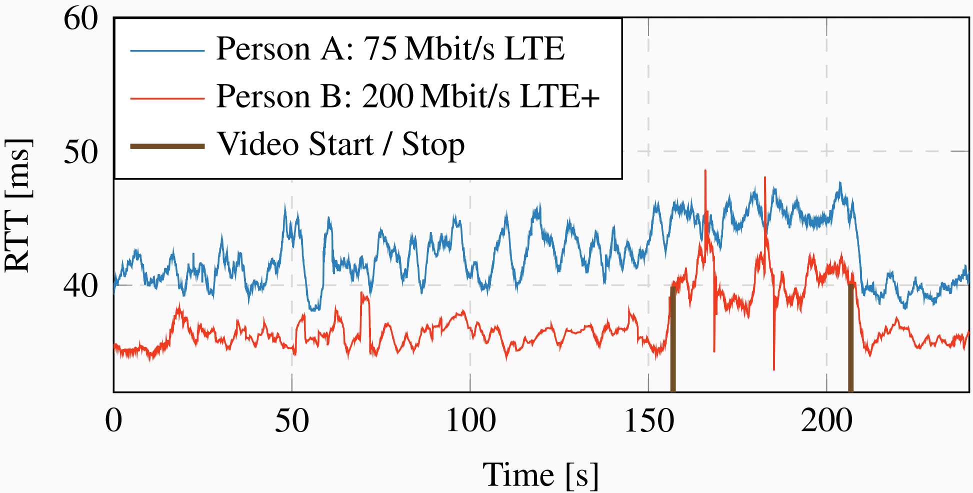
Top-100 Open-World Website Fingerprinting



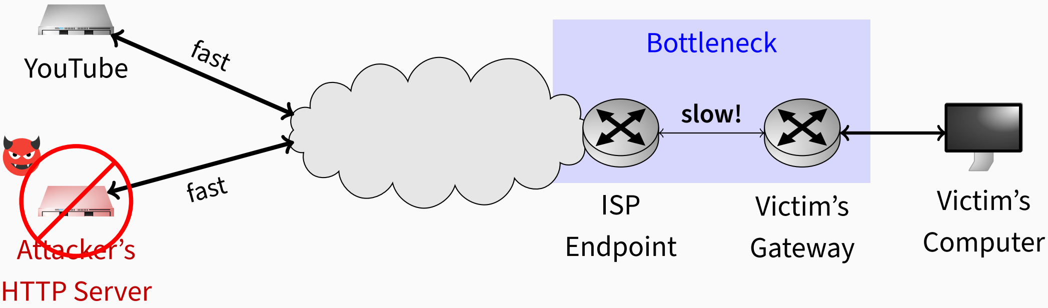
Cross-Connection Website Fingerprinting



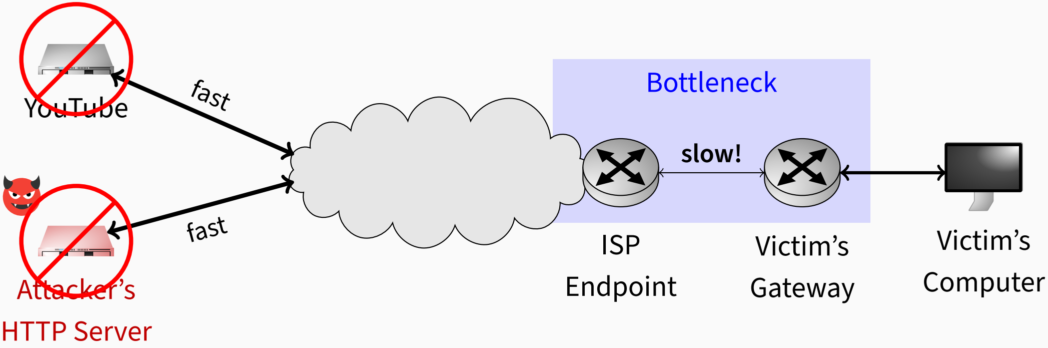
Video Call Detection



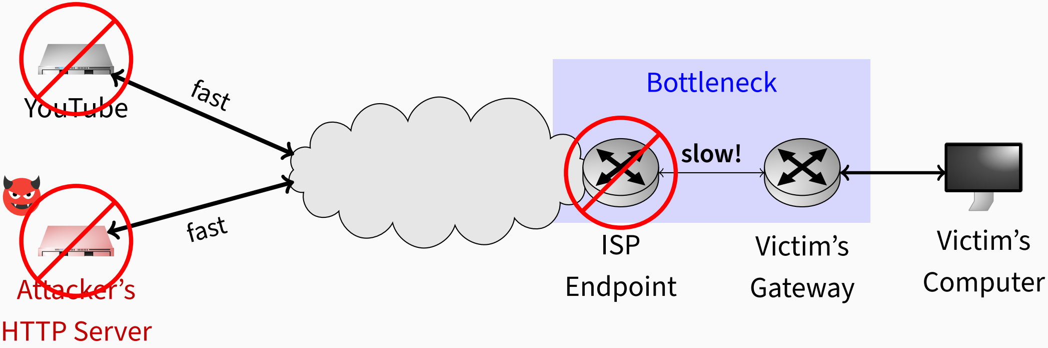
What about Mitigations?



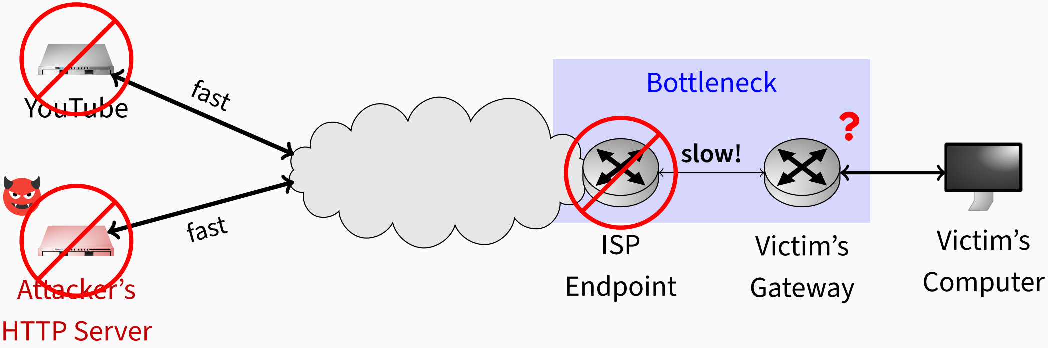
What about Mitigations?



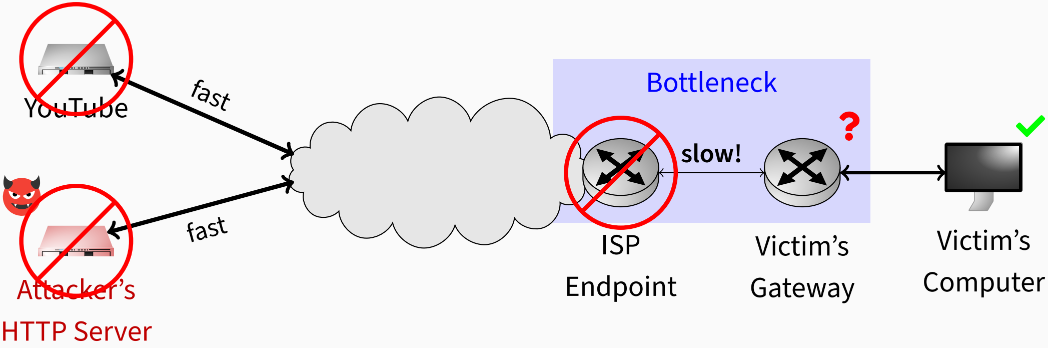
What about Mitigations?



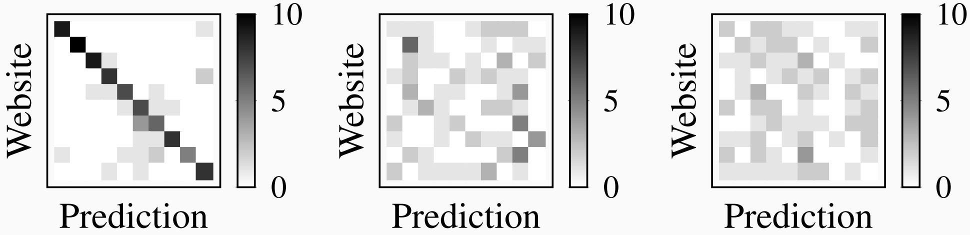
What about Mitigations?



What about Mitigations?



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

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

Take Aways

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

Take Aways

- 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

Take Aways

- 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
- Paper + Demo: `https://snailload.com`

Side-Channel Security

Chapter 7: Network Side Channels

Stefan Gast

References

[AYR15] Daniel Arp, Fabian Yamaguchi, and Konrad Rieck. **Torben: A Practical Side-Channel Attack for Deanonymizing Tor Communication.** ASIA CCS. 2015.

[Che+10] Shuo Chen et al. **Side-Channel Leaks in Web Applications: A Reality Today, a Challenge Tomorrow.** S&P. 2010.

[Gas+24] Stefan Gast et al. **SnailLoad: Exploiting Remote Network Latency Measurements without JavaScript.** USENIX Security. 2024.

[Gu+18] Jiaxi Gu et al. **Walls Have Ears: Traffic-based Side-Channel Attack in Video Streaming.** INFOCOM. 2018.

[Hin03] Andrew Hintz. **Fingerprinting Websites Using Traffic Analysis.** PET. 2003.

[ISO22] ISO/IEC. **Dynamic adaptive streaming over HTTP (DASH) (ISO/IEC 23009-1:2022).** 2022.

[LM18] Michael Lescisin and Qusay Mahmoud. **Tools for Active and Passive Network Side-Channel Detection for Web Applications.** WOOT. 2018.

[Pan+11] Andriy Panchenko et al. **Website Fingerprinting in Onion Routing Based Anonymization Networks.** WPES. 2011.

[Rim+17] Vera Rimmer et al. **Automated website fingerprinting through deep learning.** NDSS. 2017.

[RK17] Andrew Reed and Michael Kranch. **Identifying HTTPS-Protected Netflix Videos in Real-Time.** CODASPY. 2017.

[RSG98] Michael Reed, Paul Syverson, and David Goldschlag. **Anonymous Connections and Onion Routing.** Journal on Selected Areas in Communications 16.4 (1998), pp. 482–494.

[SST17] Roel Schuster, Vitaly Shmatikov, and Eran Tromer. **Beauty and the Burst: Remote Identification of Encrypted Video Streams.** USENIX Security. 2017.

[SWT01] Dawn Xiaodong Song, David Wagner, and Xuqing Tian. **Timing Analysis of Keystrokes and Timing Attacks on SSH.** USENIX Security. 2001.

[Wan+11] Xiaogang Wang et al. **A potential HTTP-based application-level attack against Tor.** Future Generation Computer Systems (2011).

[Whi+11] Andrew White et al. **Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on Fon-iks.** S&P. 2011.

[Wri+07] Charled Wright et al. **Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob?** USENIX Security. 2007.