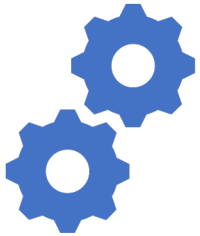
Exploiting and Mitigating Implicit Cookie-based Authentication Vulnerabilities on the Web

Tom Van Goethem

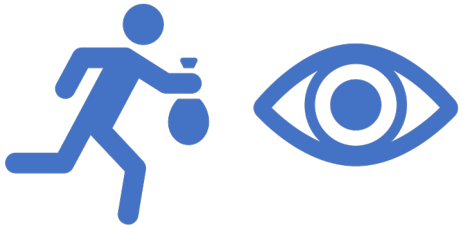
@tomvangoethem

KU LEUVEN DistriNet

Overview

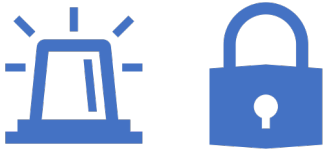


Cookies & SOP 101



Tracking & cross-site attacks

Cross-site size-exposing attacks

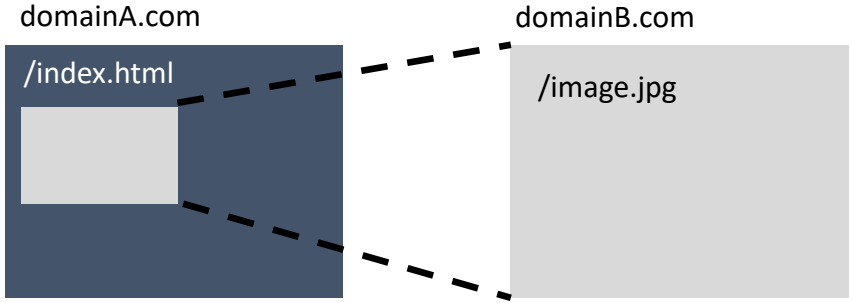


Third-party cookie policies

Comprehensive evaluation

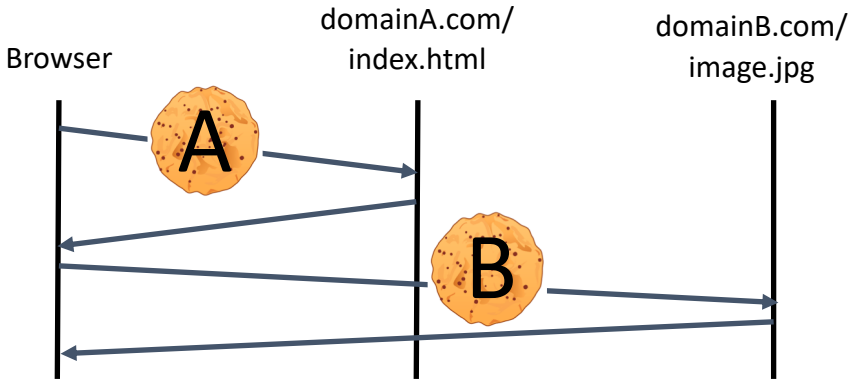
Conclusion

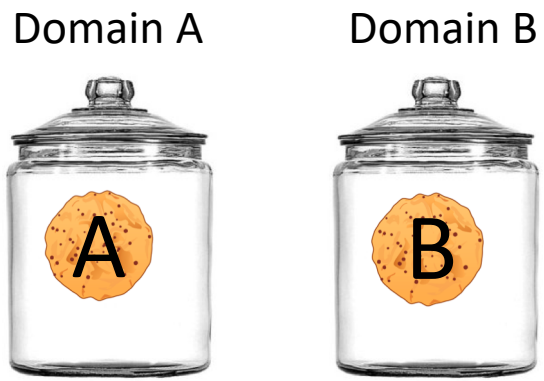
Cookie inclusion



HTTP cookies [1]

- Implicit inclusion
- Authentication / identification
- Same-Origin Policy

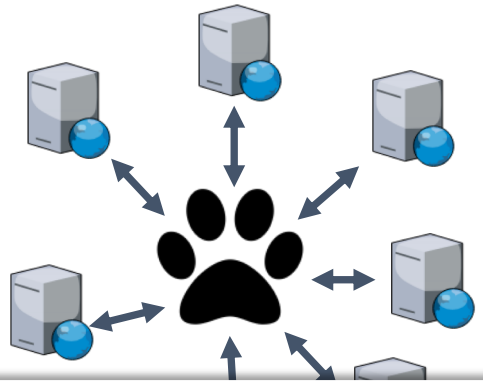




[1] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011.

Third-party tracking

Third-party Tracking



victim

cat-news.com

paw-book.com

Tracking the Trackers

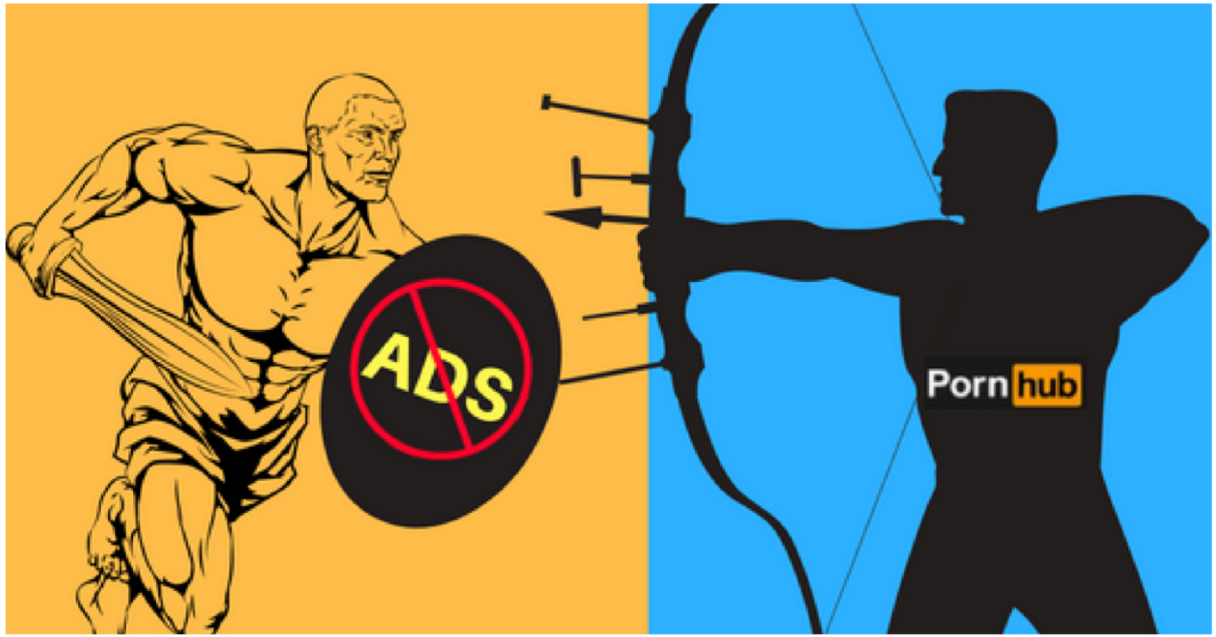
Zhonghao Yu Sam Macbeth Konark Modi

"95% of the pages visited contain 3rd party requests to potential trackers
78% attempt to transfer unsafe data"

Cliqz
Arabellastraße 23

Pornhub Bypasses Ad Blockers With WebSockets

BugReplay Follow
Nov 1, 2016 · 4 min read



Cross-site attacks

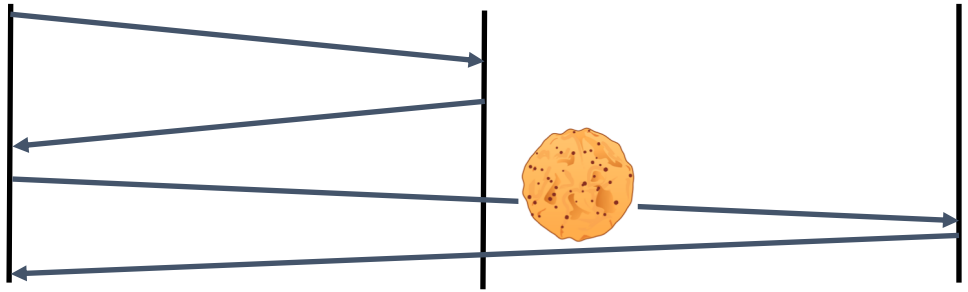
Cross-site Request Forgery (CSRF)

- Authenticated state-changing request

victim

cute-kittens.com

doggo-bank.com



```
<img src="https://doggo-bank.com/transfer.php
    ?amount=999999&recipient=catboss">
```

ZDNet

CENTRAL EUROPE

MUST READ: Meet the new Microsoft Phone, powered by Android (No Windows

could have

News Reviews How To's

APPUALS
FROM THE EXPERTS!

Follow Us

5,715 Fans

2,490 Subscribers

HELPNETSECURITY

News Features Expert Analysis Reviews Events Whitepapers Industry news Newsletters

Featured news

- UK citizens fear identity theft over other security concerns such as national security
- How science can fight insider threats
- The risk to OT networks is real, and it's dangerous for business leaders to ignore
- 66% UK SMBs believe they are being aggressively targeted by fraudsters
- Phishing attacks becoming more targeted, phishers

Zeljka Zorz, Managing Editor
October 3, 2018

Share this article

Popular TP-Link wireless home router open to remote hijacking

By concatenating a known improper authentication flaw with a newly discovered CSRF vulnerability, remote unauthenticated control over TP-Link TL-WR841N worldwide

How science can fight insider threats

How to make the CFO your best cybersecurity friend

Safeguarding hybrid-cloud infrastructures through identity privilege management

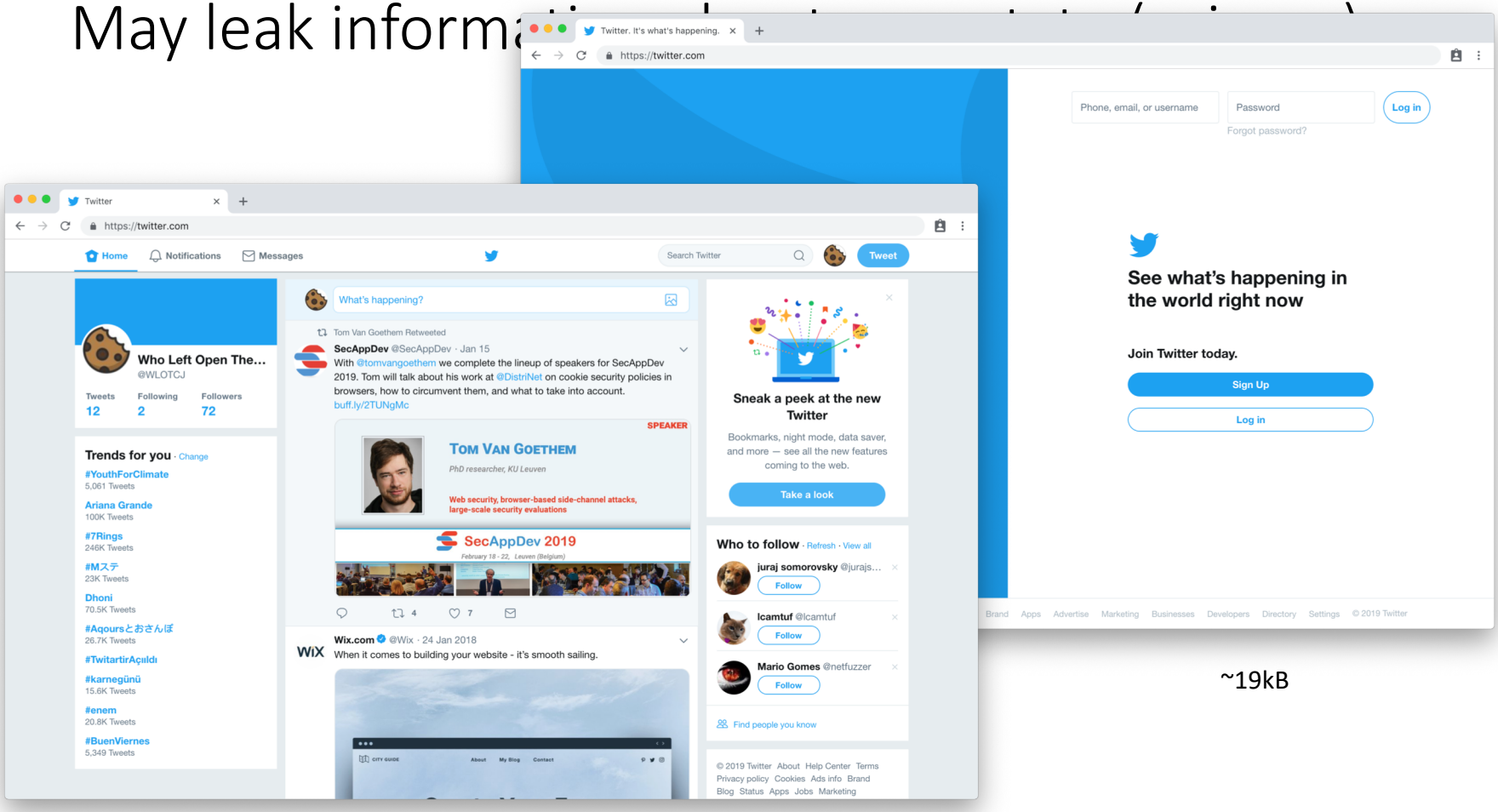
Why you should take an operational approach to risk management

Cross-site Request Forgery

- Why is this still a problem?
  - Defense (e.g. random token in request parameters) needs to be applied ubiquitously
  - Insecure by default
- How to move on from here?
  - SameSite cookies -> secure by default (if enforced correctly by the browser)

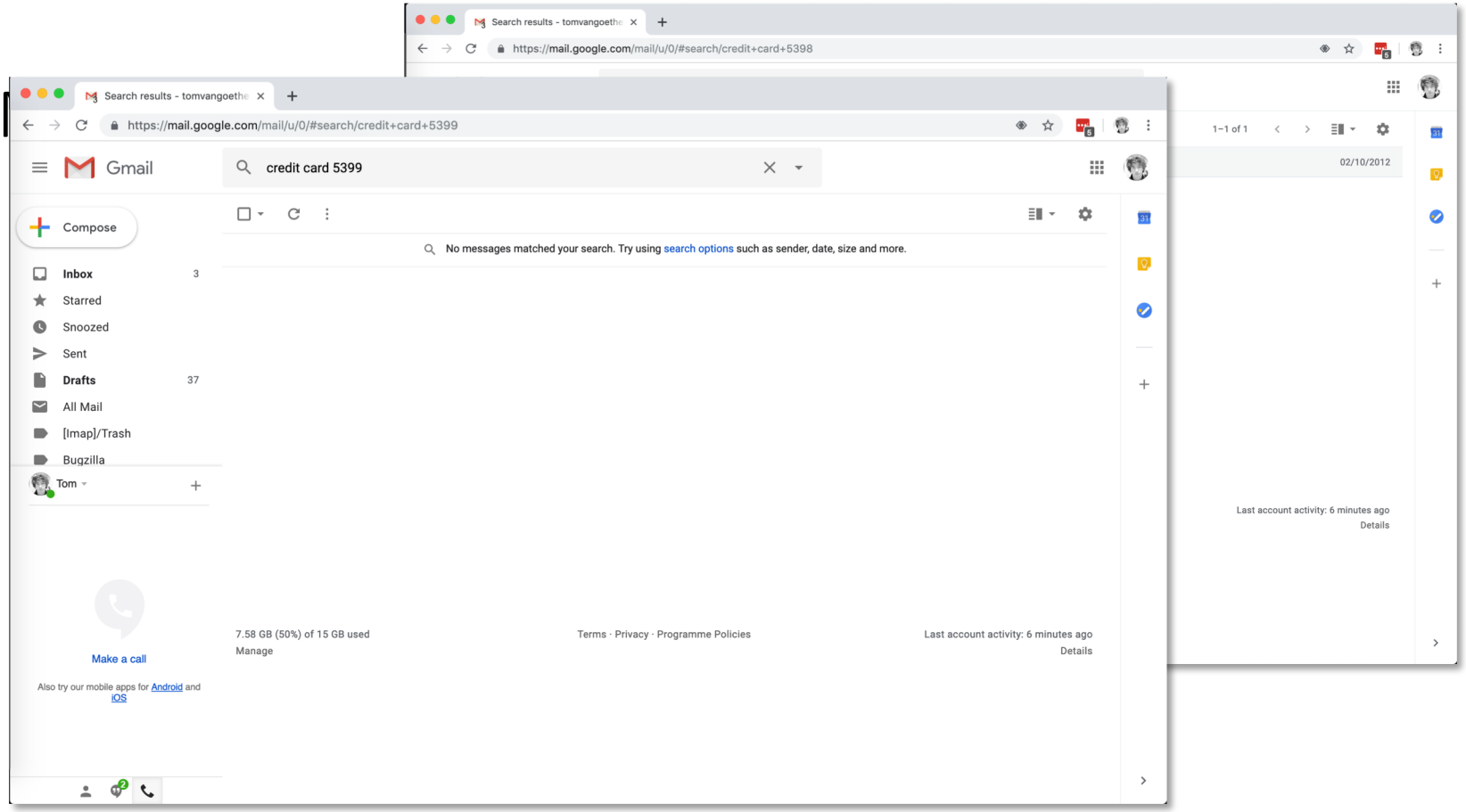
Cross-site Size-exposing Attacks

May leak informa



~19kB

~183kB



Gelernter, Nethanel, and Amir Herzberg. "Cross-site search attacks." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS). ACM, 2015.

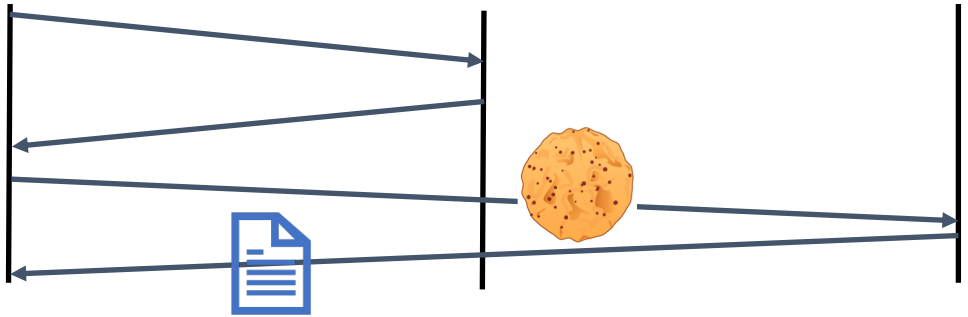
Cross-site timing attacks [1]

- State-dependent content

victim cute-kittens.com doggo-bank.com

Start timer

Stop timer

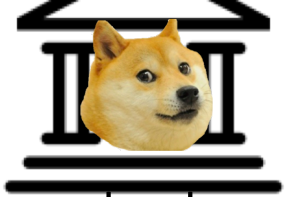


```
<img src="https://doggo-bank.com/index.hmtl">
```

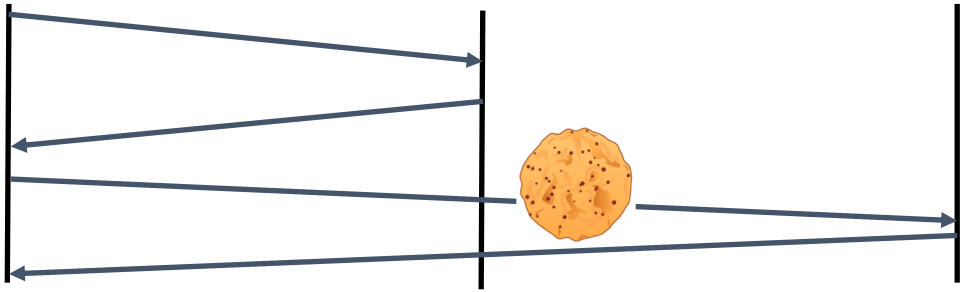
error event

[1] Bortz et al. 2007. Exposing private information by timing web applications. In Proceedings of the 16th international conference on World Wide Web (WWW '07). ACM, New York, NY, USA, 621-628.

Browser-based timing a[illegible]s [1]



victim cute-kittens.com doggo-bank.com



Start timer **suspend**

```
<video src="https://doggo-bank.com/index.hmtl">
```

Stop timer **error**

→ Abuse of firing events during parsing process

- **suspend** when fetched
- **error** on fail

[1] Van Goethem et al. The Clock is Still Ticking: Timing Attacks in the Modern Web. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15). ACM, New York, NY, USA, 1382-1393.

Video Parsing Attack

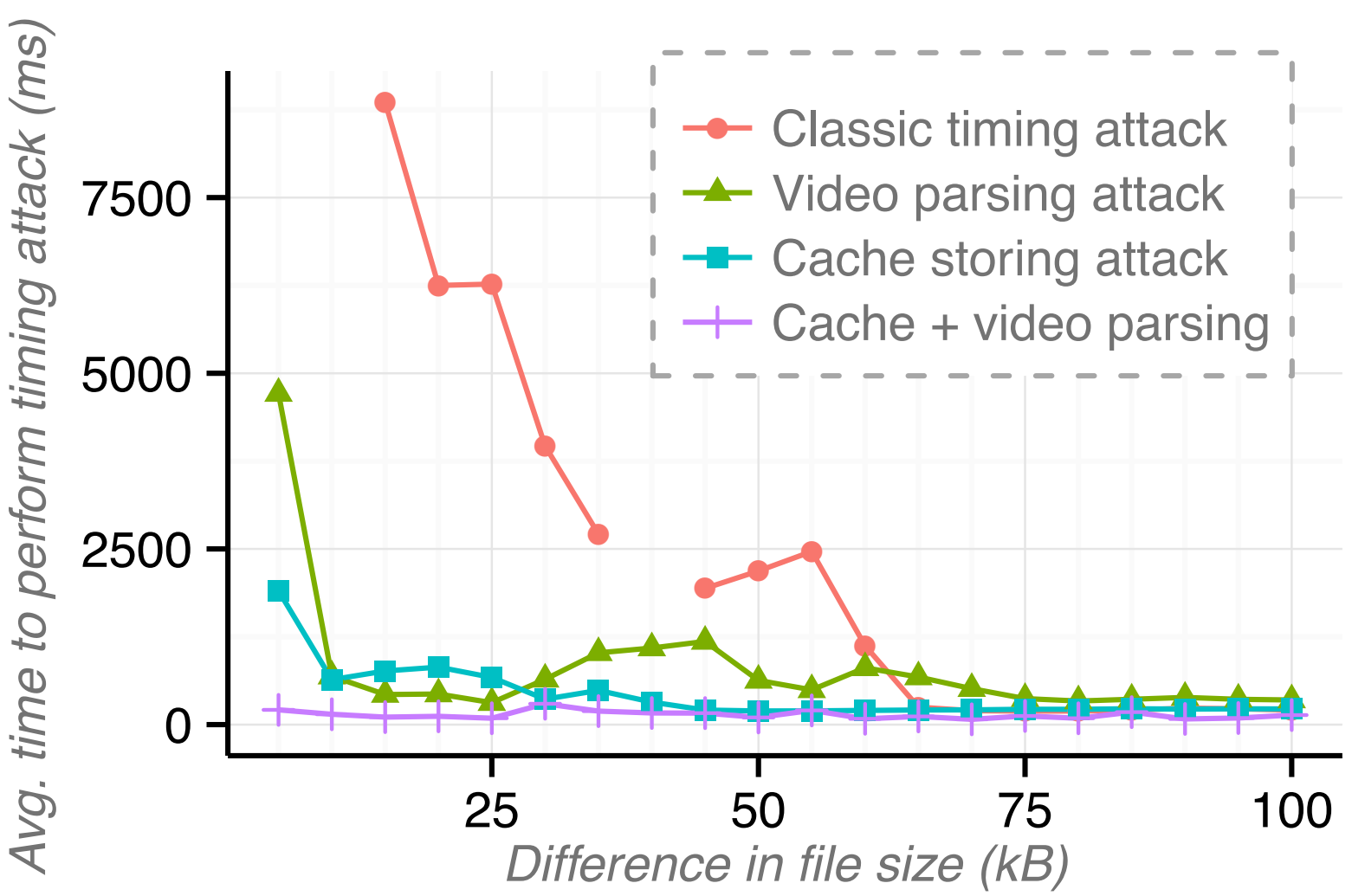
```
let video = document.createElement('video');

// suspend => download complete
video.addEventListener('suspend',function(){
    start = window.performance.now();
});

// error => parsing complete
video.addEventListener('error',function(){
    end = window.performance.now();
});

video.src = 'https://example.org/resource';
```

appcache.manifest

```
CACHE MANIFEST
CACHE:
https://example.org/resource
NETWORK:
*
```

Cache Storage Attack

```
let url = 'https://example.org/resource';
let opts = {credentials: "include", mode: "no-cors"};
let request = new Request(url, opts);
let bogusReq = new Request('/bogus');
fetch(request).then(function(resp) {
    // Resource download complete
    start = window.performance.now();
    return cache.put(bogusReq, resp.clone())
}).then(function() {
    // Resource stored in cache
    end = window.performance.now();
});
```


Avg. time to perform timing attack (ms)
7500
5000
2500
0
25
50
75
100
Difference in file size (kB)
Classic timing attack
Video parsing attack
Cache storing attack
Cache + video parsing

Browser-based Size Leaking

- Can differentiate resource that differ few KB
- Video parsing mechanisms already patched is several browsers
  - New features may cause new side-channels (e.g. SRI, image parsing, …)
- Real-world attacks can be improved by using response inflation
  - One result is repeated many times → difference in response size is artificially enlarged
- Attacks discovered in 2016; bug hunters starting to leverage techniques

XS-Searching Google's bug tracker to find out vulnerable source code

Or how side-channel timing attacks aren't that impractical

Luan Herrera Follow

Nov 19, 2018 · 6 min read

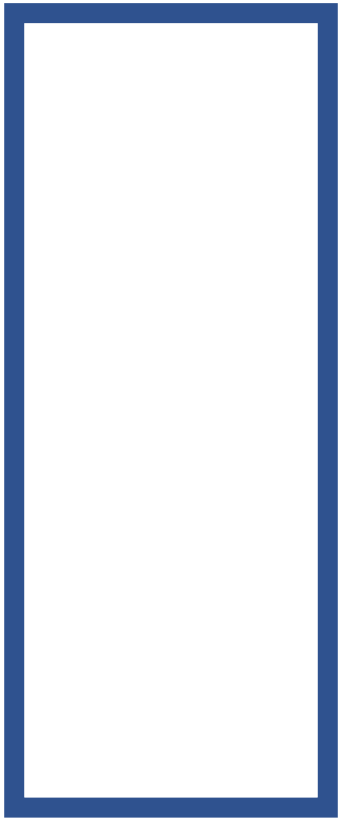
Monorail is an open-source issue tracker used by many "Chromium-orbiting" projects, including Monorail itself. Other projects include Angle, PDFium, Gerrit, V8, and the Alliance for Open Media. It is also used by Project Zero, Google's 0-day bug-finding team.

This article is a detailed explanation of how I could have exploited Google's Monorail issue tracker to leak sensitive information (vulnerable source code

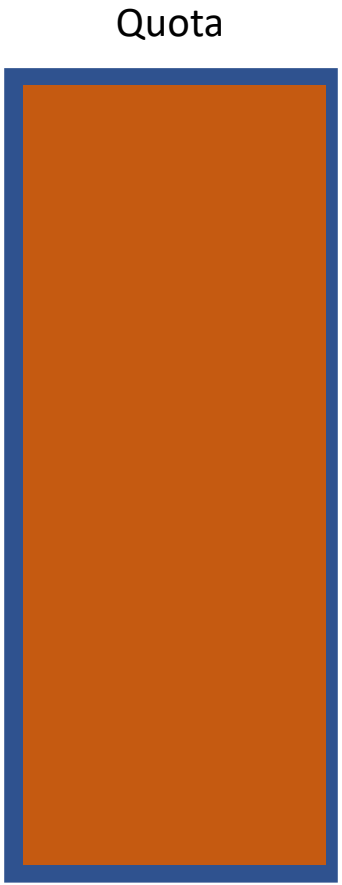
Abusing Storage Quota

- Each site (eTLD+1) has a specific quota
  - IndexedDB, localStorage, …
  - Cross origin resources (!!!)
- When quota is reached, any attempt to store more is blocked
- Can be used to determine **exact size** of cross-origin resource
- Exact size --> defenses against response inflation do not work

Quota

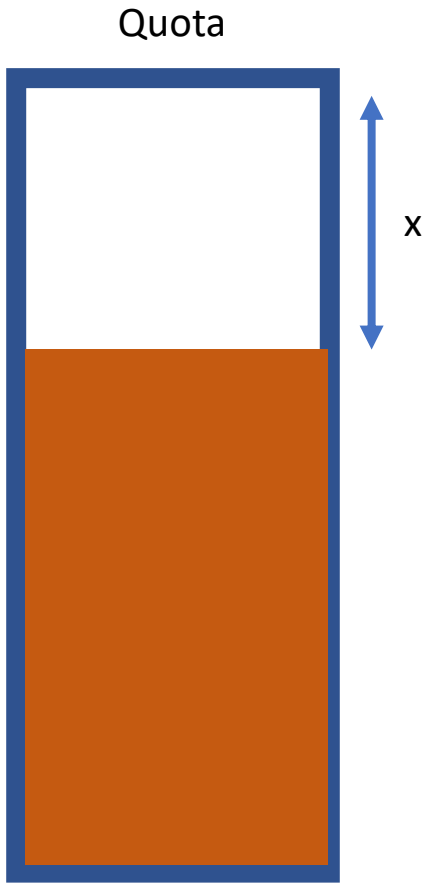


Step 1: fill



Step 1: fill

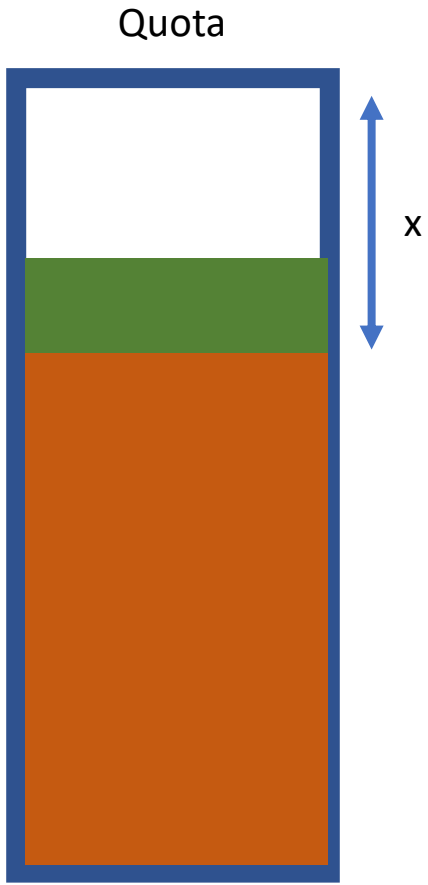
Step 2: remove x



Step 1: fill

Step 2: remove x

Step 3: store resource

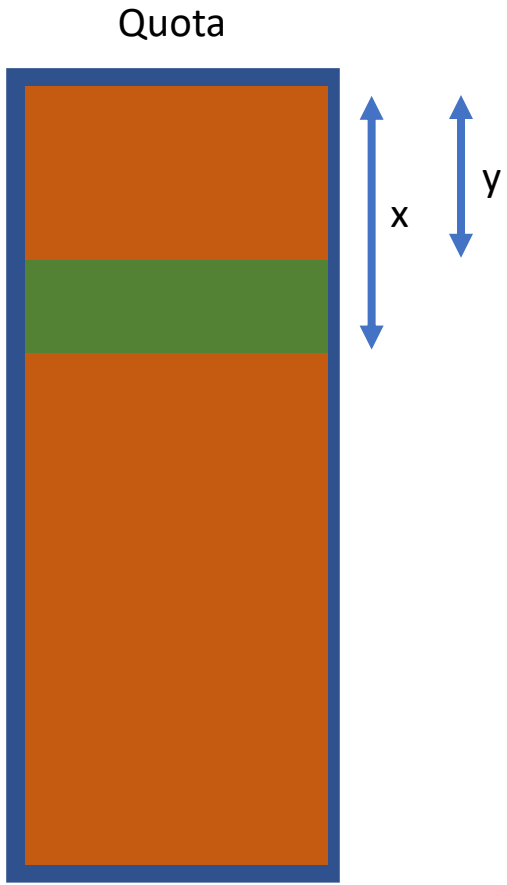


Step 1: fill

Step 2: remove x

Step 3: store resource

Step 4: fill



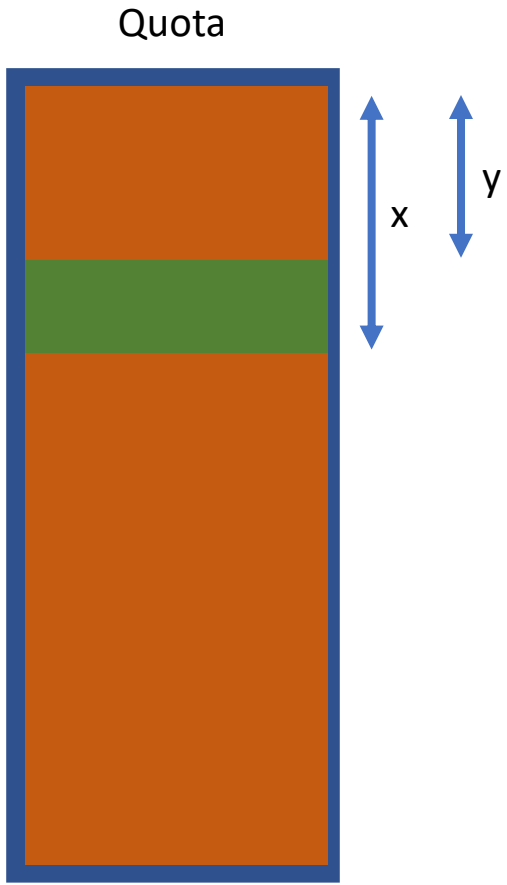
Step 1: fill

Step 2: remove x

Step 3: store resource

Step 4: fill

Step 5: x - y = PROFIT



Quota Management API

- Developers may want to know how many bytes are available/used
- Quota API returns “estimate”
  - In reality, the estimate provided exact number of bytes
- Attack becomes super easy
  - x = getEstimate(); store(crossOriginResource); y = getEstimate(); size = y - x;

Storage/Quota API status

- Fixes have been deployed
  - For every stored cross-origin resource, a random number of bytes (approx. 7MB in Chrome) count towards the quota
- Low-impact solution, highly effective
  - No performance impact; small usability impact (for sites that store many cross-orgin resources)
  - Very few attack scenarios left
    - Maybe abuse global quota & trigger website to store resource same-origin (highly unlikely)

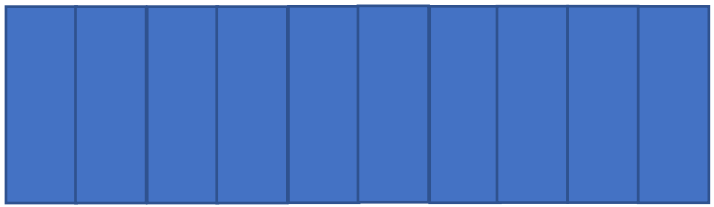

CROSS-SITE ATTACKS
LIGHTYEAR
CROSS-SITE ATTACKS EVERYWHERE

HEIST

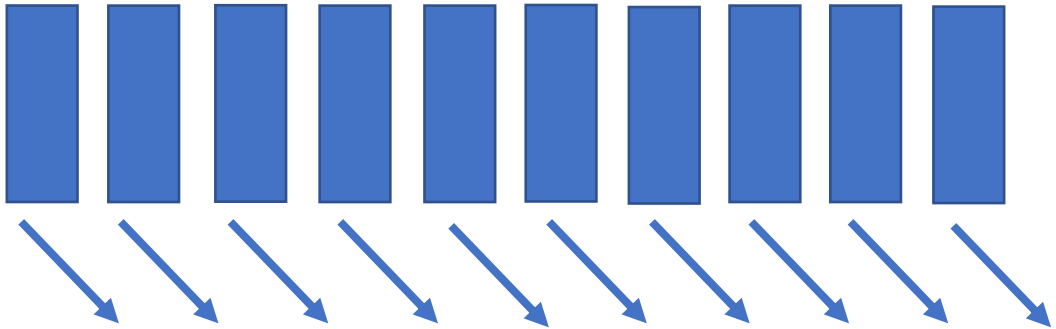
(HTTP Encrypted Information can be Stolen through TCP Windows)

- Determine **exact** response size (compressed)
- 1 TCP window = 10 TCP packets = 14480 bytes of data
- 2nd TCP window can only start after ACK (--> additional round-trip)
- Response fits in 1 TCP window --> 1 RTT, otherwise 2+ RTTs
- Use side-channel to detect when headers are received
  - fetch() promise resolves
- Use side-channel to detect when full response is received
  - Cache API store + read
- Timing difference < 5ms --> 1 TCP window, otherwise 2 TCP windows

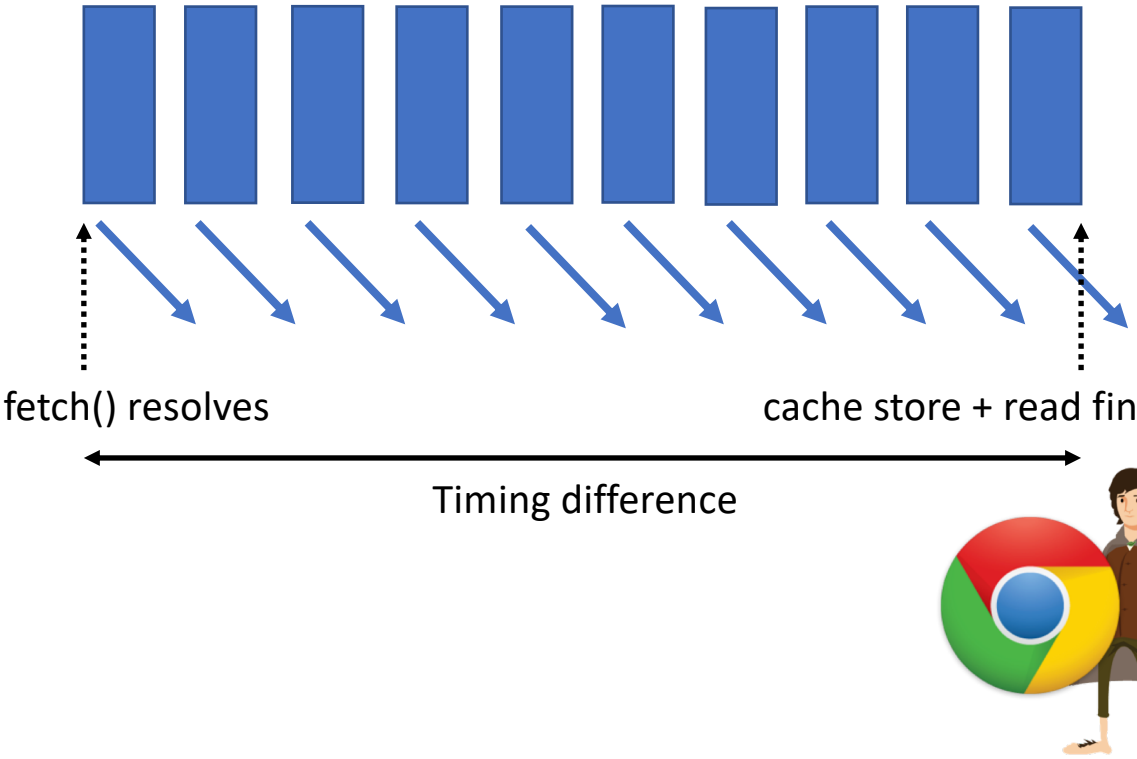
Response (14480 bytes)



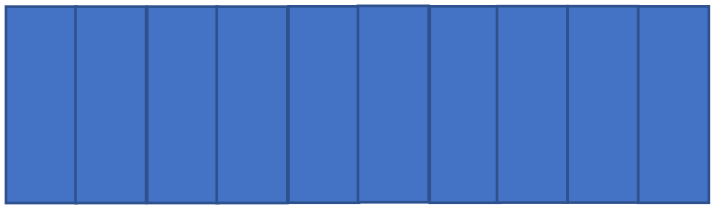
1st TCP window



1st TCP window

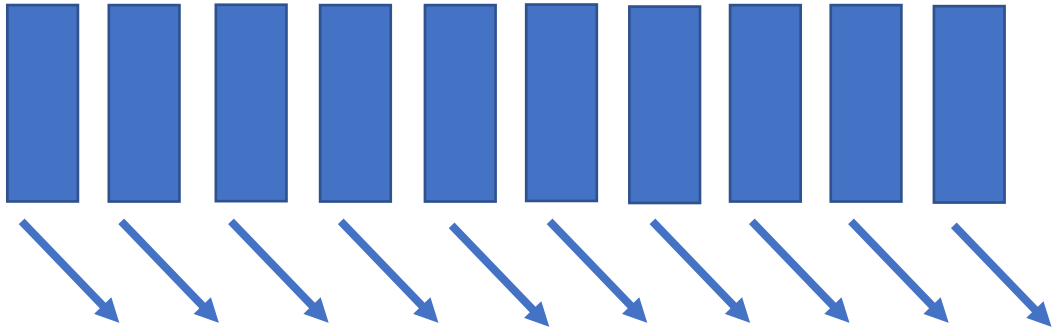


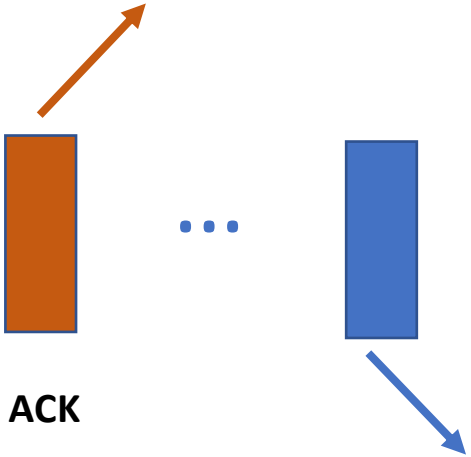
Response (14481 bytes)

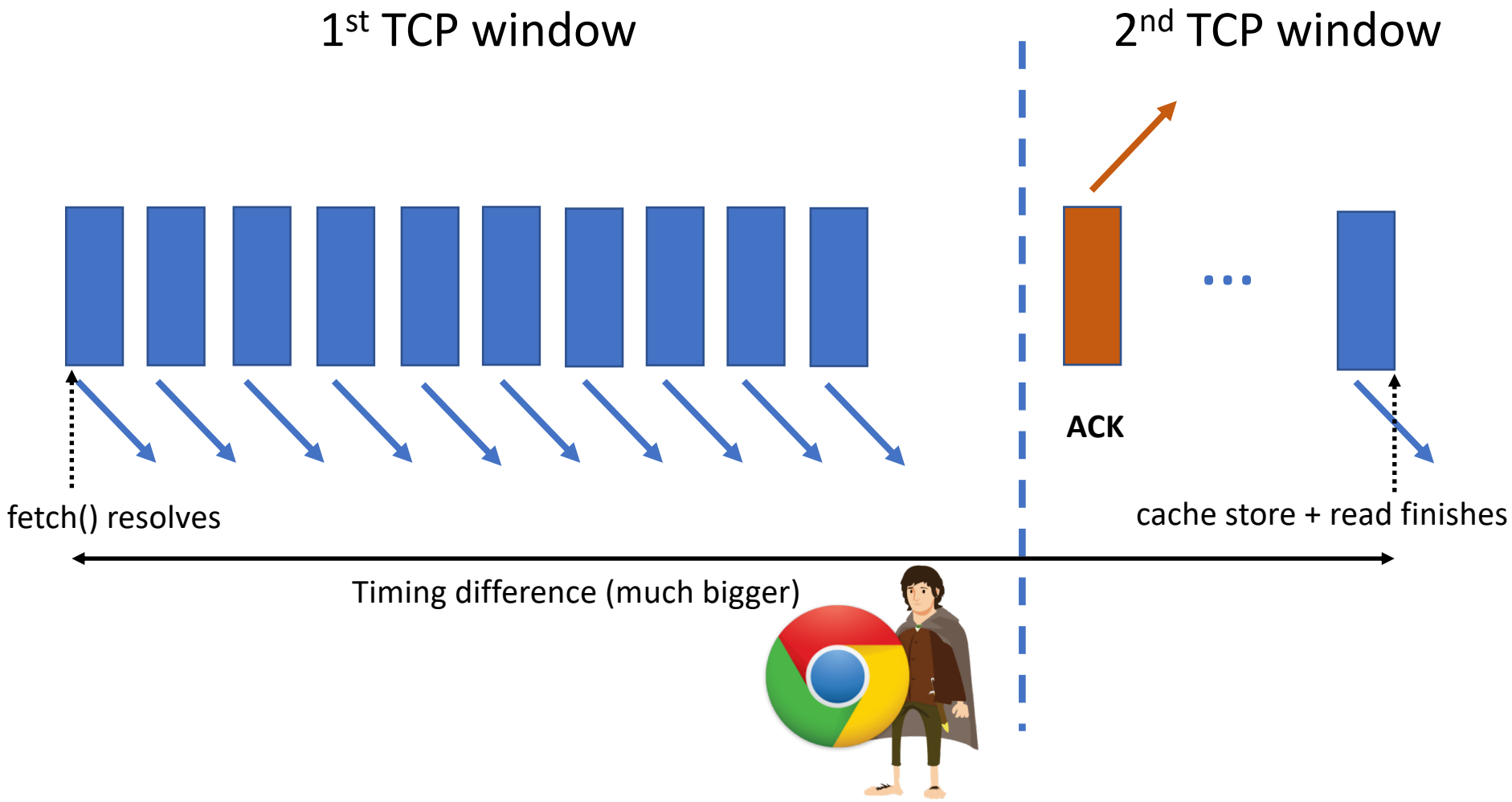


1st TCP window

2nd TCP window






1st TCP window
2nd TCP window
ACK
fetch() resolves
cache store + read finishes
Timing difference (much bigger)

HEIST

- Important prerequisite: reflection of request in response
  - Needed to align on TCP window size
- Exact size is known **after** compression
  - Allows for BREACH-like attack

Hello *$_GET['name']*, your secret value is SWAG_MEISTER

?name=Tom

gzip(Hello Tom, your secret value is SWAG_MEISTER)

==> Hello Tom, your secret value is SWAG_MEISTER

?name=SWAG

gzip(Hello SWAG, your secret value is SWAG_MEISTER)

==> Hello SWAG, your secret value is @-27,4_MEISTER

?name=SWAGx

gzip(Hello SWAGx, your secret value is SWAG_MEISTER)

==> Hello SWAGx, you secret value is @-27,4_MEISTER

--> 42 bytes

?name=SWAG_

gzip(Hello SWAG_, your secret value is SWAG_MEISTER)

==> Hello SWAG_, you secret value is @-28,5MEISTER

--> 41 bytes

HEIST

- Can be used to extract cross-origin secrets (CSRF tokens)
- Defense: disable compression for sensitive content
  - https://blog.cloudflare.com/a-solution-to-compression-oracles-on-the-web/
  - Not widely deployed, requires regex to know what is sensitive
- Defense: refresh tokens after N requests
  - Can be tricky + what about other sensitive content?
- Large-scale impact: to be explored

Defenses

Same-site cookie [1]

- In-depth defense against cross-site attacks

- Cookie with extra attribute ‘SameSite’
  - SameSite=strict → NO CROSS-SITE REQUESTS!
  - SameSite=lax → exceptions: top-level GET, prerender

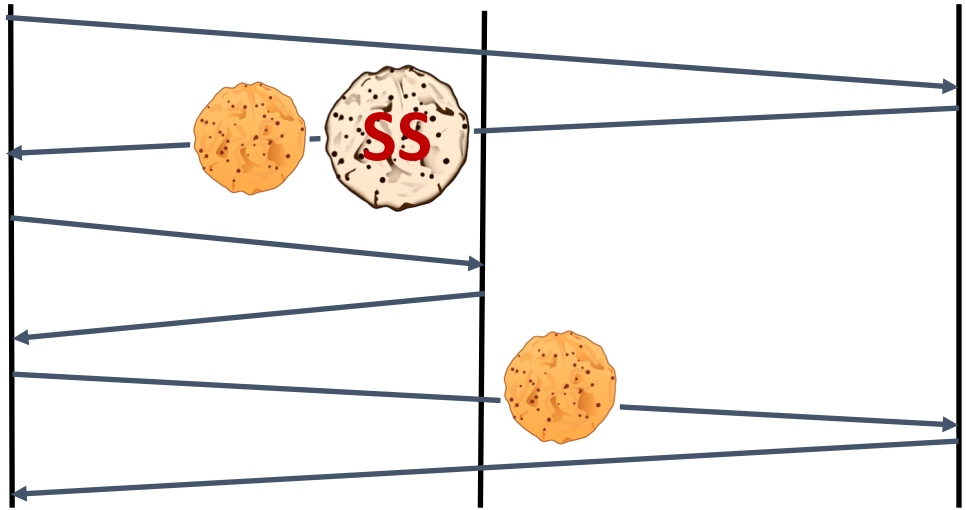
- Adoption by websites is rather slow
  - Interesting blog: Dropbox’s use case [2]

[1] West, M., Goodwin, M. Same-site cookies. Internet- Draft draft-ietf-httpbis-cookie-same-site-00, IETF Secretariat, June 2016.
[2] https://blogs.dropbox.com/tech/2017/03/preventing-cross-site-attacks-using-same-site-cookies/

Use of same-site cookies

- against cross-site attacks

doggo-bank.com victim cute-kittens.com doggo-bank.com




```
Set-Cookie: auth=ekSd2lksq090pQDs; SameSite=strict
```

What about privacy?

- Built-in browser options
  - Block third-party cookies
  - Firefox Tracking Protection
  - Opera Ad Blocker
  - Safari Intelligent Tracking Prevention

- Extensions
  - Ad blocking
  - Privacy protection

Client-side defense mechanisms

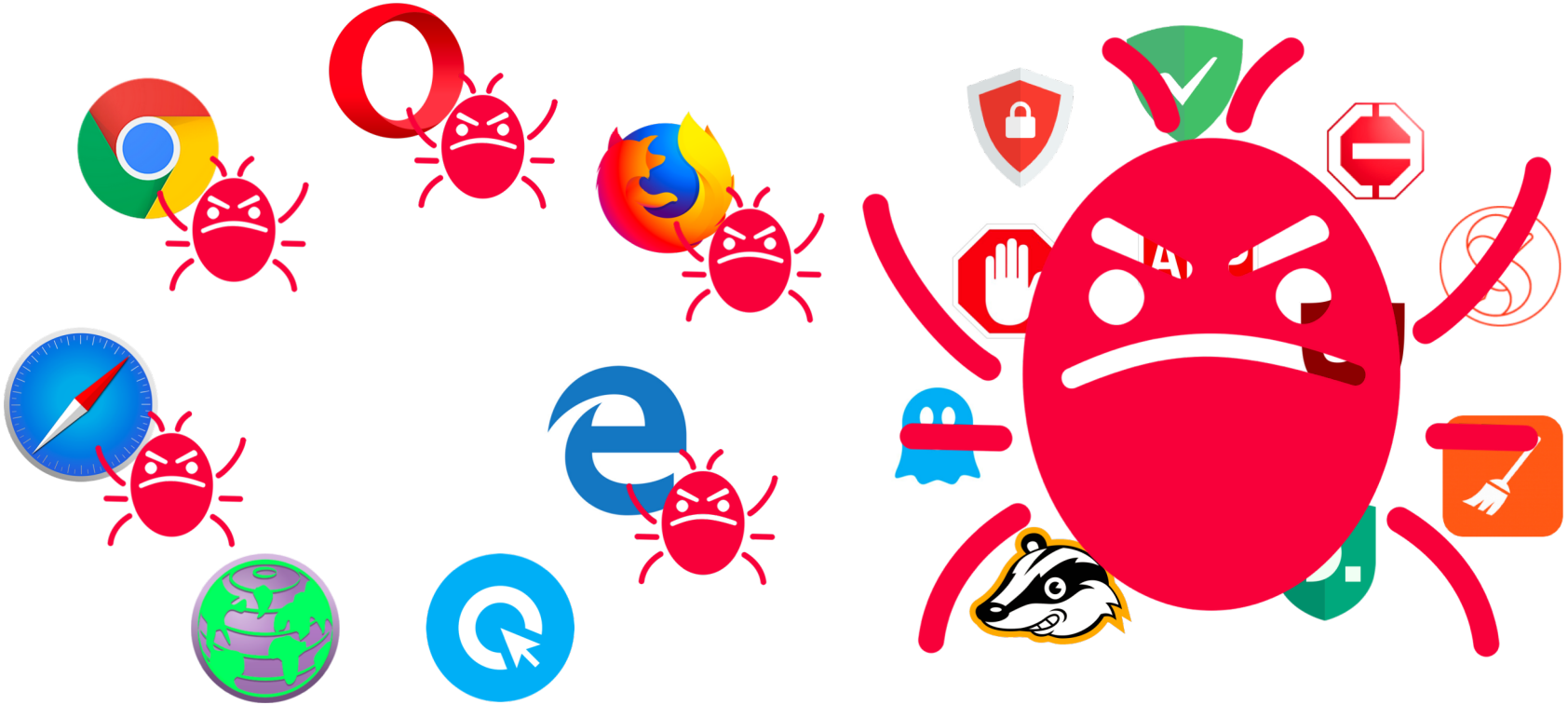
Security measures only work when they are **consistently** and **universally** applied!

Why evaluate third-party cookie policies?

- Browsers are known to exhibit inconsistent behavior
  - Interference from different standards
  - Unintended side-effects by code modification

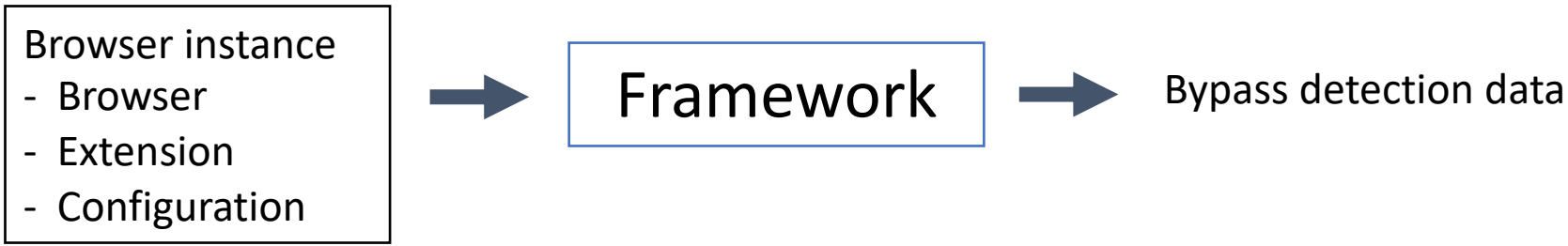
- Saturated market of extensions
  - No clear quantification of quality

Automated evaluation of effectiveness



Black box approach

- Browsers consist of millions of lines of code
  - Source code not always available
- Many extensions



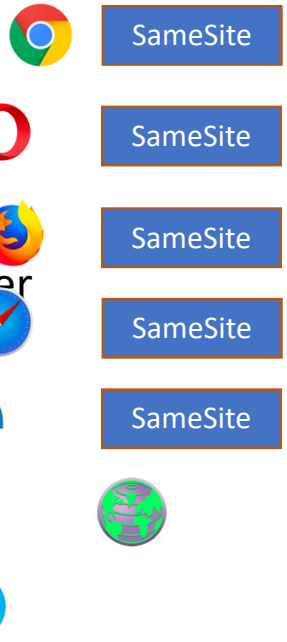
Initiating cross-site requests

- AppCache API
  - Caching cross-site pages
- HTML-tags
  - <script>, <img>, <link>, etc.
- Headers
  - Link, CSP headers

- Redirects
- JavaScript
  - Fetch, EventSource API, etc.
- PDF JS
  - sendForm()
- ServiceWorker API

Overview

- Browsers
  - Chrome
  - Opera
  - Firefox
  - Safari
  - Edge
  - Tor Browser
  - Cliqz



- Extensions
  - Ad blocking (31)

...

  - Tracking protection (15)

...

	AppCache	HTML	Headers	Redirects	PDF JS	JavaScript	SW
Chrome 63	●	●	●	●	●	●	●
- Block third-party cookies	◐	◐	◐	●	●	◐	◐
Opera 51	●	●	●	●	●	●	●
- Block third-party cookies*	◐	◐	◐	●	●	◐	◐
- Ad Blocker	●	●	○	●	○	●	●
Firefox 57	●	●	●	●	○	●	●
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
- Tracking Protection	●	●	●	●	○	●	●
Safari 11	○†	◐	○	●	○	◐	N/A
- No Intelligent Tracking Prevention	●†	●	○	●	○	●	N/A
- Block third-party cookies‡	●†	●	◐	●	○	●	N/A
Edge 40	●	●	◐	●	○	●	N/A
- Block third-party cookies	●	●	◐	●	○	●	N/A
Cliqz 1.17*	◐	●	◐	●	○	◐	◐
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
Tor Browser 7	○	◐	◐	●	○	◐	N/A

●: request with cookies ◐: request without cookies ○: no request

* Secure cookies were omitted in all requests.

† Safari does not permit cross-domain caching over https (only over http).

‡ Safari 10.1.2

	AppCache	HTML	Headers	Redirects	PDF JS	JavaScript	SW
Chrome 63	●	●	●	●	●	●	●
- Block third-party cookies	◐	◐	◐	●	●	◐	◐
Opera 51	●	●	●	●	●	●	●
- Block third-party cookies*	◐	◐	◐	●	●	◐	◐
- Ad Blocker	●	●	○	●	○	●	●
Firefox 57	●	●	●	●	○	●	●
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
- Tracking Protection	●	●	●	●	○	●	●
Safari 11	○†	◐	○	●	○	◐	N/A
- No Intelligent Tracking Prevention	●†	●	○	●	○	●	N/A
- Block third-party cookies‡	●†	●	◐	●	○	●	N/A
Edge 40	●	●	◐	●	○	●	N/A
- Block third-party cookies	●	●	◐	●	○	●	N/A
Cliqz 1.17*	◐	●	◐	●	○	◐	◐
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
Tor Browser 7	○	◐	◐	●	○	◐	N/A

●: request with cookies ◐: request without cookies ○: no request

* Secure cookies were omitted in all requests.

† Safari does not permit cross-domain caching over https (only over http).

‡ Safari 10.1.2

	AppCache	HTML	Headers	Redirects	PDF JS	JavaScript	SW
Chrome 63	●	●	●	●	●	●	●
- Block third-party cookies	◐	◐	◐	●	●	◐	◐
Opera 51	●	●	●	●	●	●	●
- Block third-party cookies*	◐	◐	◐	●	●	◐	◐
- Ad Blocker	●	●	○	●	○	●	●
Firefox 57	●	●	●	●	○	●	●
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
- Tracking Protection	●	●	●	●	○	●	●
Safari 10	○†	◐	○	●	○	◐	N/A
- No Intelligent Tracking Prevention	●†	●	○	●	○	●	N/A
- Block third-party cookies‡	●†	●	◐	●	○	●	N/A
Edge 40	●	●	◐	●	○	●	N/A
- Block third-party cookies	●	●	◐	●	○	●	N/A
Cliqz 1.17*	◐	●	◐	●	○	◐	◐
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
Tor Browser 7	○	◐	◐	●	○	◐	N/A

●: request with cookies ◐: request without cookies ○: no request

* Secure cookies were omitted in all requests.

† Safari does not permit cross-domain caching over https (only over http).

‡ Safari 10.1.2

	AppCache	HTML	Headers	Redirects	PDF JS	JavaScript	SW
Chrome 63	●	●	●	●	●	●	●
- Block third-party cookies	◐	◐	◐	●	●	◐	◐
Opera 51	●	●	●	●	●	●	●
- Block third-party cookies*	◐	◐	◐	●	●	◐	◐
- Ad Blocker	●	●	○	●	○	●	●
Firefox 57	●	●	●	●	○	●	●
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
- Tracking Protection	●	●	●	●	○	●	●
Safari 11	○†	◐	○	●	○	◐	N/A
- No Intelligent Tracking Prevention	●‡	●	○	●	○	●	N/A
- Block third-party cookies‡	●†	●	◐	●	○	●	N/A
Edge 40	●	●	◐	●	○	●	N/A
- Block third-party cookies	●	●	◐	●	○	●	N/A
Cliqz 1.17*	◐	●	◐	●	○	◐	◐
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
Tor Browser 7	○	◐	◐	●	○	◐	N/A

●: request with cookies ◐: request without cookies ○: no request

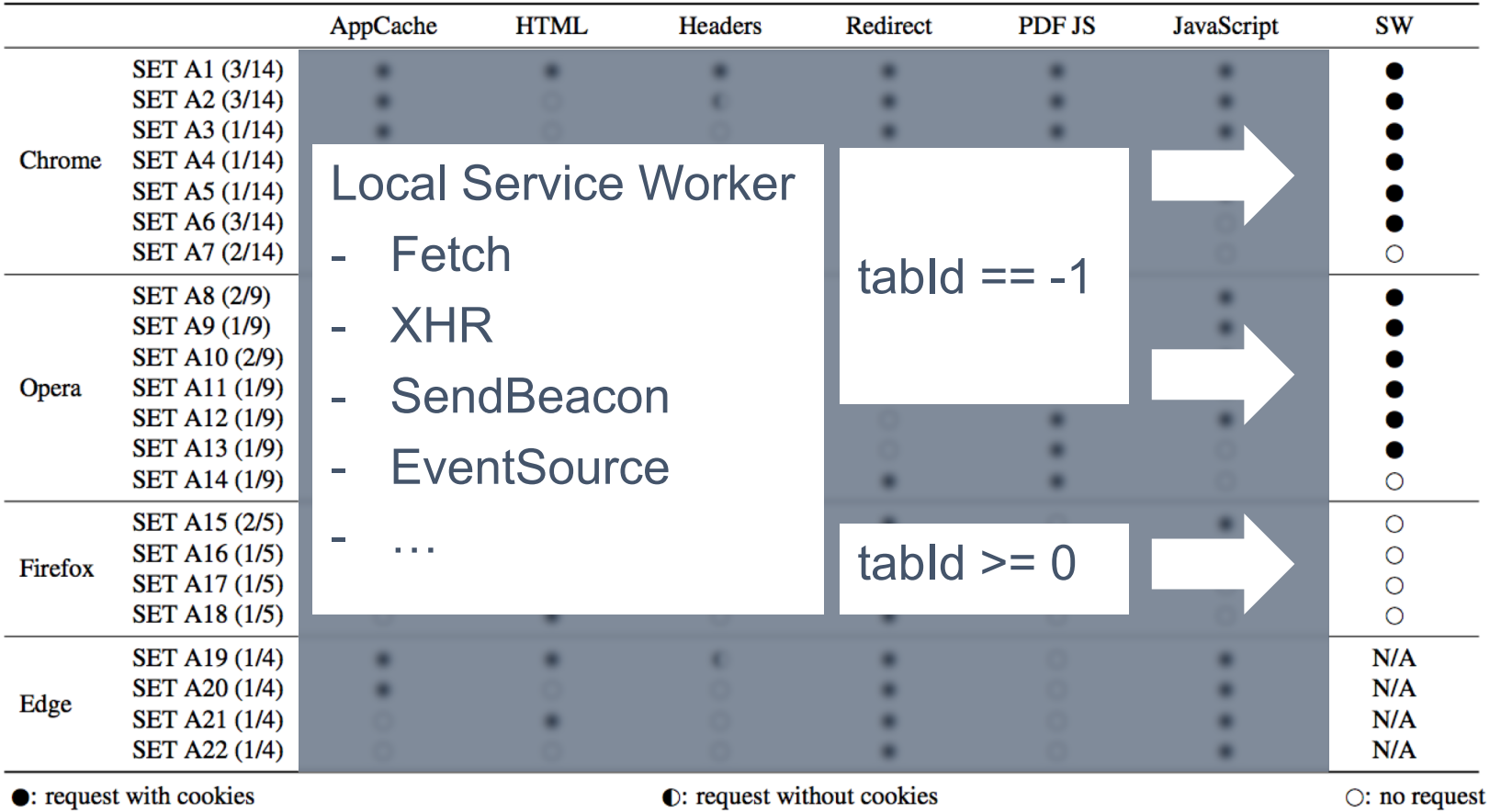
* Secure cookies were omitted in all requests.

† Safari does not permit cross-domain caching over https (only over http).

‡ Safari 10.1.2

		AppCache	HTML	Headers	Redirect	PDF JS	JavaScript	SW
Chrome	SET A1 (3/14)	●	●	●	●	●	●	●
	SET A2 (3/14)	●	○	◐	●	●	●	●
	SET A3 (1/14)	●	○	○	●	●	●	●
	SET A4 (1/14)	●	○	○	●	●	○	●
	SET A5 (1/14)	●	○	○	○	●	●	●
	SET A6 (3/14)	●	○	○	○	●	○	●
	SET A7 (2/14)	○	○	○	●	●	○	○
Opera	SET A8 (2/9)	●	●	●	●	●	●	●
	SET A9 (1/9)	●	○	◐	●	●	●	●
	SET A10 (2/9)	●	○	○	●	●	●	●
	SET A11 (1/9)	●	○	○	●	●	○	●
	SET A12 (1/9)	●	○	○	○	●	●	●
	SET A13 (1/9)	●	○	○	○	●	○	●
	SET A14 (1/9)	○	○	○	●	●	○	○
Firefox	SET A15 (2/5)	●	●	◐	●	○	●	○
	SET A16 (1/5)	●	●	○	●	○	○	○
	SET A17 (1/5)	●	●	○	○	○	○	○
	SET A18 (1/5)	○	●	○	●	○	○	○
Edge	SET A19 (1/4)	●	●	◐	●	○	●	N/A
	SET A20 (1/4)	●	○	○	●	○	●	N/A
	SET A21 (1/4)	○	●	○	●	○	●	N/A
	SET A22 (1/4)	○	○	○	●	○	●	N/A

●: request with cookies ◐: request without cookies ○: no request

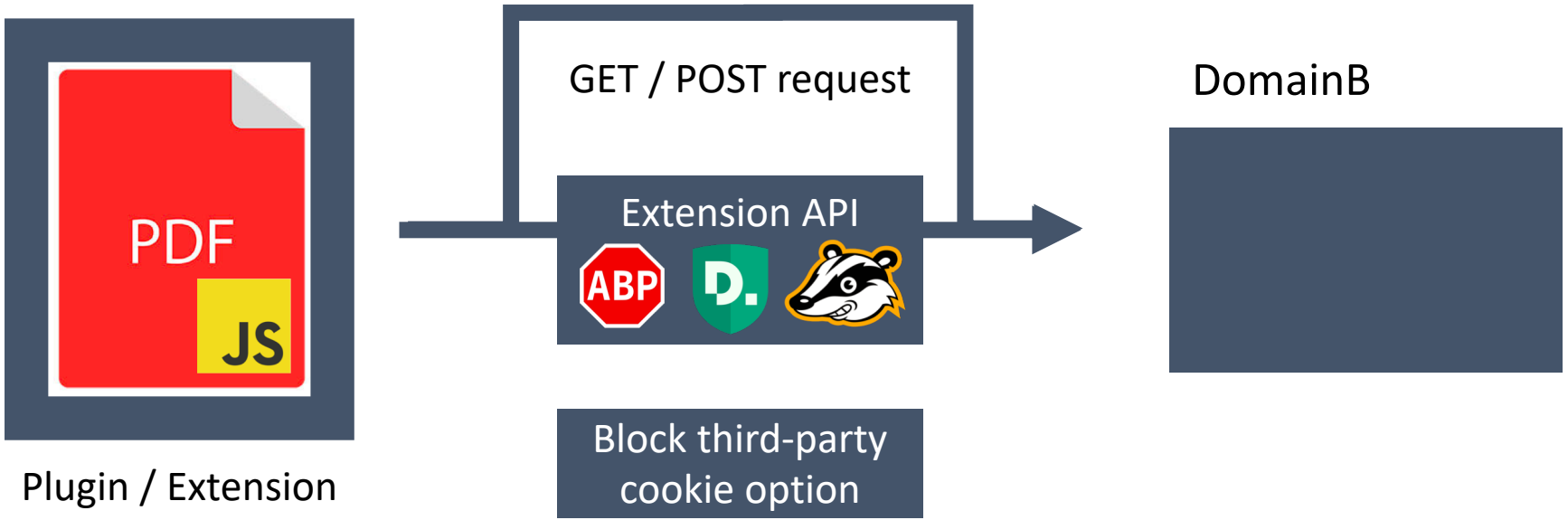

AppCache
HTML
Headers
Redirect
PDF JS
JavaScript
SW
Chrome
SET A1 (3/14)
SET A2 (3/14)
SET A3 (1/14)
SET A4 (1/14)
SET A5 (1/14)
SET A6 (3/14)
SET A7 (2/14)
Opera
SET A8 (2/9)
SET A9 (1/9)
SET A10 (2/9)
SET A11 (1/9)
SET A12 (1/9)
SET A13 (1/9)
SET A14 (1/9)
Firefox
SET A15 (2/5)
SET A16 (1/5)
SET A17 (1/5)
SET A18 (1/5)
Edge
SET A19 (1/4)
SET A20 (1/4)
SET A21 (1/4)
SET A22 (1/4)
N/A
Local Service Worker
- Fetch
- XHR
- SendBeacon
- EventSource
- …
tabId == -1
tabId >= 0
●: request with cookies
◐: request without cookies
○: no request

		AppCache	HTML	Headers	Redirect	PDF JS	JavaScript	SW
Chrome	SET A1 (3/14)	●	●	●	●	●	●	●
	SET A2 (3/14)	●	○	◐	●	●	●	●
	SET A3 (1/14)	●	○	○	●	●	●	●
	SET A4 (1/14)	●	○	○	●	●	○	●
	SET A5 (1/14)	●	○	○	○	●	●	●
	SET A6 (3/14)	●	○	○	○	●	○	●
	SET A7 (2/14)	○	○	○	●	●	○	○
Opera	SET A8 (2/9)	●	●	●	●	●	●	●
	SET A9 (1/9)	●	○	◐	●	●	●	●
	SET A10 (2/9)	●	○	○	●	●	●	●
	SET A11 (1/9)	●	○	○	●	●	○	●
	SET A12 (1/9)	●	○	○	○	●	●	●
	SET A13 (1/9)	●	○	○	○	●	○	●
	SET A14 (1/9)	○	○	○	●	●	○	○
Firefox	SET A15 (2/5)	●	●	◐	●	○	●	○
	SET A16 (1/5)	●	●	○	●	○	○	○
	SET A17 (1/5)	●	●	○	○	○	○	○
	SET A18 (1/5)	○	●	○	●	○	○	○
Edge	SET A19 (1/4)	●	●	◐	●	○	●	N/A
	SET A20 (1/4)	●	○	○	●	○	●	N/A
	SET A21 (1/4)	○	●	○	●	○	●	N/A
	SET A22 (1/4)	○	○	○	●	○	●	N/A

●: request with cookies ◐: request without cookies ○: no request

PDFium design flaw

- Chrome and Opera



Extensions

- No extension managed to block all third-party cookies to blacklisted domains
- Insufficient API
  - PDF JS for Chromium, but also Firefox favicon (HTML tags)
- Unclear API
  - No clear distinction for browser background requests
- Common mistakes
  - Insufficient permissions to intercept certain requests

Same-site cookie policy

- Chrome and Opera: prerender functionality
  - Both lax and strict included in cross-site request
- Edge
  - Lax bypasses: WebSocket API, <embed>, <object>
  - Strict bypasses: WebSocket API, redirects
- Firefox and Safari: no bugs detected

Evaluation of the framework

- Completeness and novelty
- Distributed crawler setup
  - Interception of headless Chrome network traffic (using linux network namespaces)
  - Analysis of intercepted HTTP requests
- Alexa Top 10,000 websites
  - Up to 20 pages on each website
  - 160,059 pages visited

Conclusion

Conclusion

- Browsers are complex
  - May lead to various vulnerabilities
  - 3 different techniques to obtain cross-origin resource size
- Built-in browser policies can be bypassed
  - Same-site cookie, third-party cookie policies
  - Advanced options (e.g. Opera AdBlocker, Firefox Tracking Protection)
- All adblocking and privacy extensions can be bypassed
  - Due to extension API provided by browsers
  - Due to common mistakes by extension developers

Thank you!

Twitter: @tomvangoethem

Email: tom.vangoethem@cs.kuleuven.be

KU LEUVEN DistriNet