



GLOBAL APPSEC
TEL AVIV
2019

Who left open the cookie jar?

Tom Van Goethem





OpenEMR security flaws could have exposed millions of patient records

Over 20 severe bugs were found using only manual methods by a single cybersecurity group.



By Charlie Osborne for Zero Day | August 8, 2018 -- 10:40 GMT (11:40 BST) | Topic: Security

30

The health records of millions of patients worldwide were potentially left open to attack by a slew of critical vulnerabilities uncovered by a single cybersecurity team.

OpenEMR is a popular, open-source software solution for the management of millions of electronic patient records worldwide. However, the software, until recently, also contained over 20 severe security issues.

MORE SECURITY NEWS

This is how government spyware StrongPity uses security researchers' work against them

Facebook approaches major cybersecurity firms, acquisition goals in mind

RECOMMENDED FOR YOU

Guide to Antivirus (AV) Replacement: What You Need to Know Before Replacing Your Current AV Solution

White Papers provided by CrowdStrike

DOWNLOAD NOW

MORE FROM CHARLIE OSBORNE

Security Most enterprise vulnerabilities

PHP security flaws could have



Home > News > Security > CSRF Vulnerability in phpMyAdmin 4.7.x Lets Attackers Delete Records through malicious URLs

Security

CSRF Vulnerability in phpMyAdmin 4.7.x Lets Attackers Delete Records through malicious URLs

0 Comments 1 minute read

By Zainab Imran • September 6, 2018

A Cross-Site Request Forgery (CSRF) vulnerability has been found in the phpMyAdmin version 4.7.x (before version 4.7.7) through which malicious attackers are able to perform fundamental database operations by tricking users into clicking on maliciously crafted URLs. This vulnerability has been combined under the CVE assigned to previous CSRF vulnerabilities in phpMyAdmin as

Follow Us

5,715 Fans

2,490 Subscribers

Trending



MediaTek's Helio P70 Upgraded is Kind of a Disappointment

By Sikandar Mahmood 3 hours ago



Hitman 2 'Untouchable' Trailer Shows off 7 Different Environments

By Farhan Ali 6 hours ago



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

TP-Link security flaws could have

Follow Us

5,715



2,490 Subscribers

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 love Microsoft the most



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 attackers can obtain control over TP-Link TL-WR841N, a popular wireless router 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

Vulnerable controllers could allow attackers to manipulate...



(IN)SECURE

Tracking the Trackers

Zhonghao Yu
Cliqz
Arabellastraße 23
Munich, Germany
zhonghao@cliqz.com

Sam Macbeth
Cliqz
Arabellastraße 23
Munich, Germany
sam@cliqz.com

Konark Modi
Cliqz
Arabellastraße 23
Munich, Germany
konarkm@cliqz.com

Josep M. Pujol
Cliqz
Arabellastraße 23
Munich, Germany
josep@cliqz.com

ABSTRACT

Online tracking poses a serious privacy challenge that has drawn significant attention in both academia and industry. Existing approaches for preventing user tracking, based on curated blocklists, suffer from limited coverage and coarse-grained resolution for classification, rely on exceptions that impact sites' functionality and appearance, and require significant manual maintenance. In this paper, we propose a

wards protecting their privacy online. According to [30] ad-blocking usage grew by 70% in 2014, culminating in 41% of people aged between 18-29 using an ad-blocker. This figure is consistent with the results of the empirical evaluation of 200,000 users in Germany presented in this paper.

Any person browsing the Web today is under constant monitoring from entities who track the navigation patterns of users. Previous work [19] reported that 99% of the top 200

“95% of the pages visited contain 3rd party requests to potential trackers”

Client-side cookie policies

- › Defend against the perils of third-party cookies
- › Built-in browser options
 - ›› Block third-party cookies
 - ›› Same-site cookies
 - ›› Firefox Tracking Protection
 - ›› Opera Ad Blocker
 - ›› Safari Intelligent Tracking Prevention
- › Browser extensions
 - ›› Ad blockers
 - ›› Privacy protection

Client-side cookie policies

- › Defend against the perils of third-party cookies

- › Built-in browser options

- ›› Block third-party cookies

- ›› Same-site cookies

- ›› Firefox Tracking Protection

- ›› Opera ad-blocker

- ›› Safari Intelligent Tracking Prevention

- › Browser extensions

- ›› Ad blockers

- ›› Privacy protection



**NEEDS TO BE CORRECTLY
ENFORCED BY THE BROWSER**





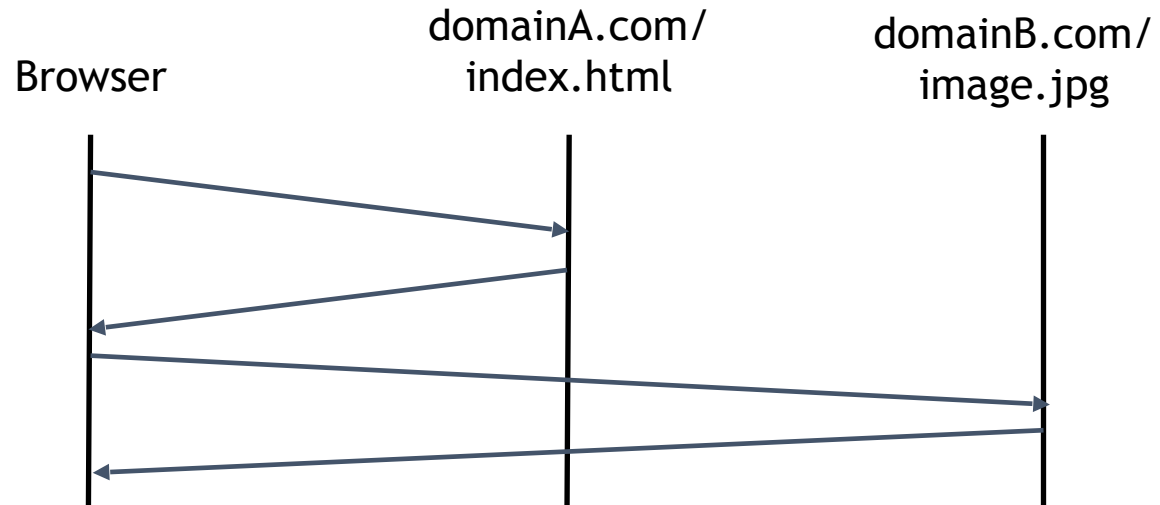
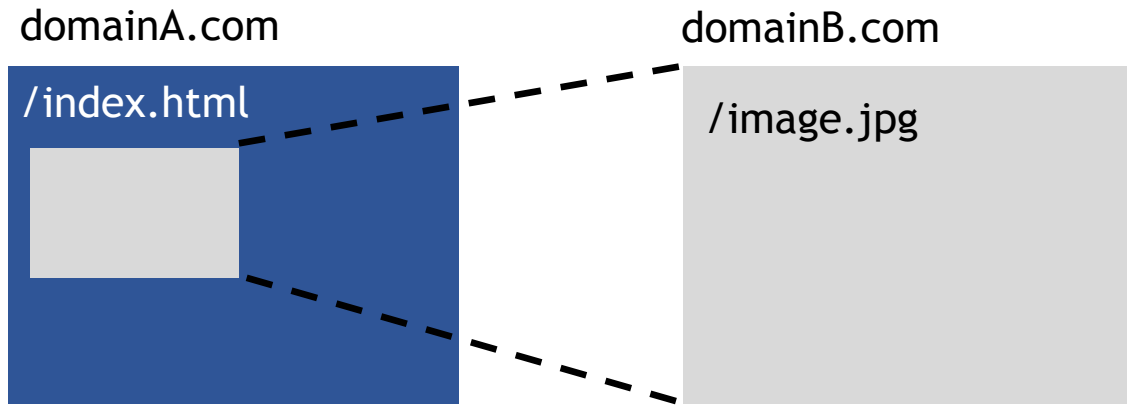




Outline

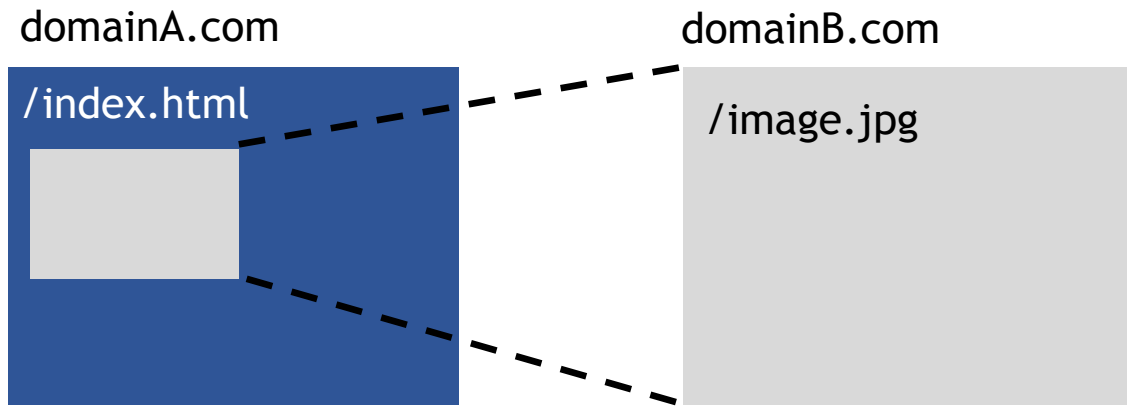
- › Background & motivation
- › Cookie policy testing framework
- › Results
- › Evaluating browser security/privacy policies
- › Conclusion

Web Fundamentals

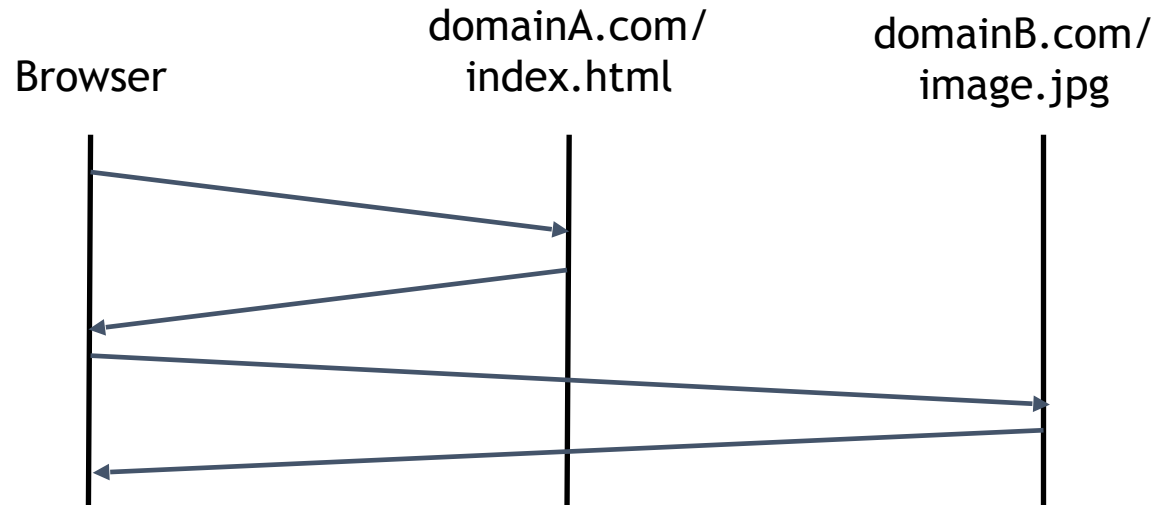


Web Fundamentals

HTTP cookies

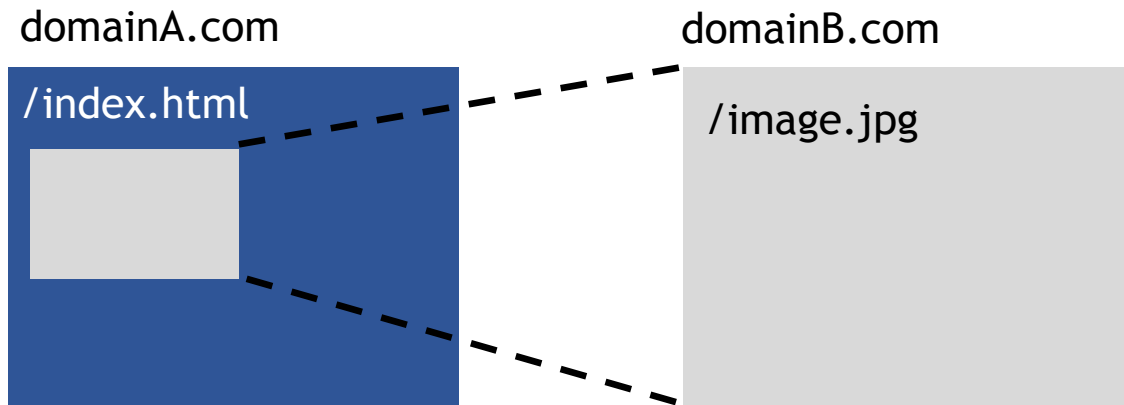


- › Implicit inclusion
- › Authentication / identification

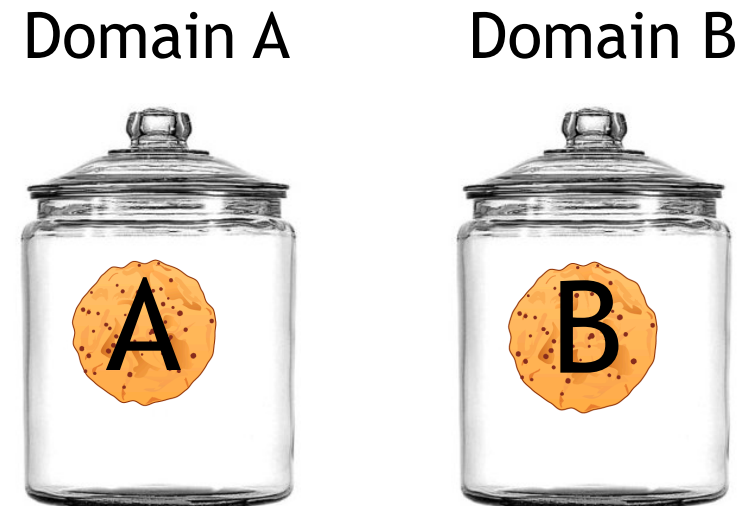
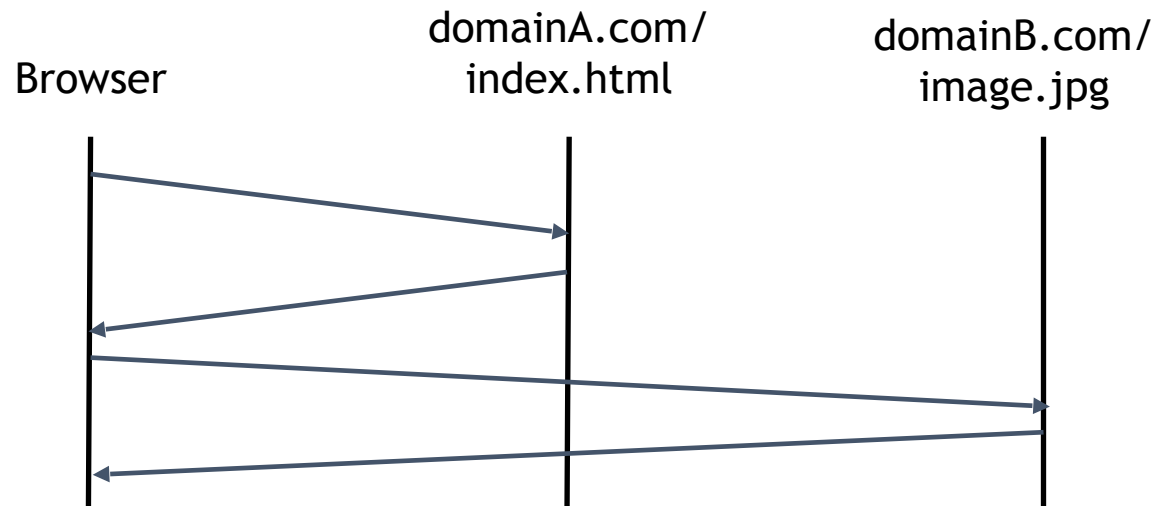


Web Fundamentals

HTTP cookies

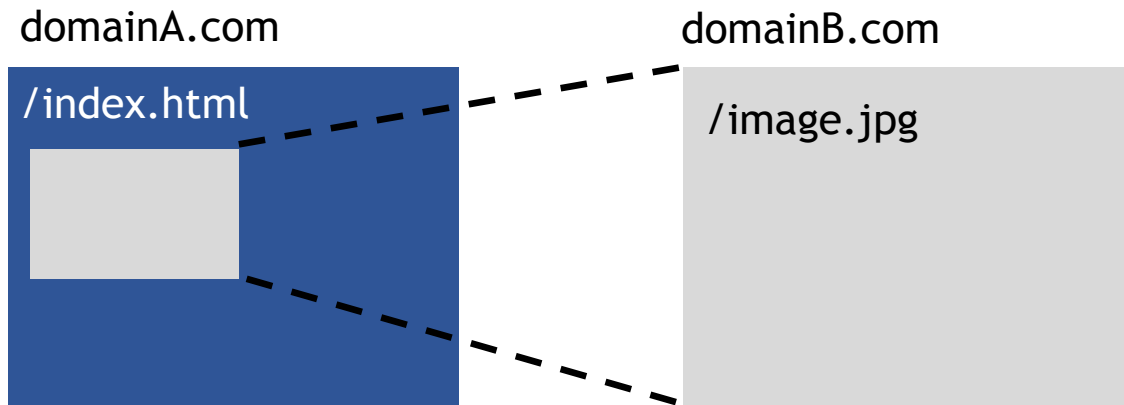


- › Implicit inclusion
- › Authentication / identification
- › Same-Origin Policy

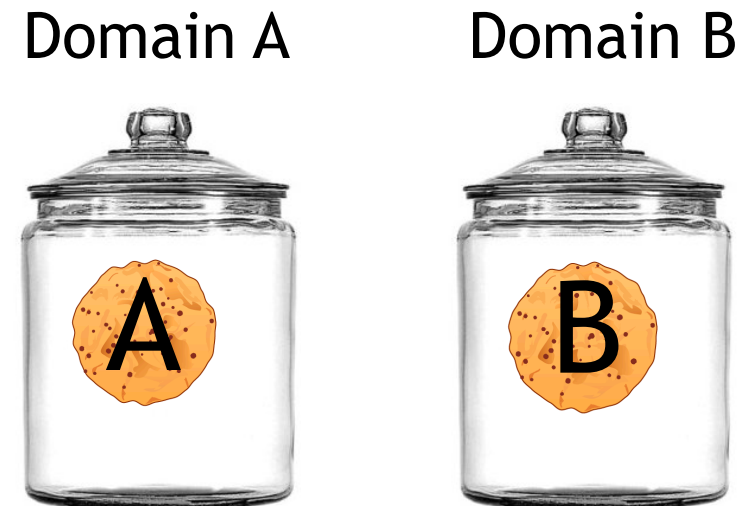
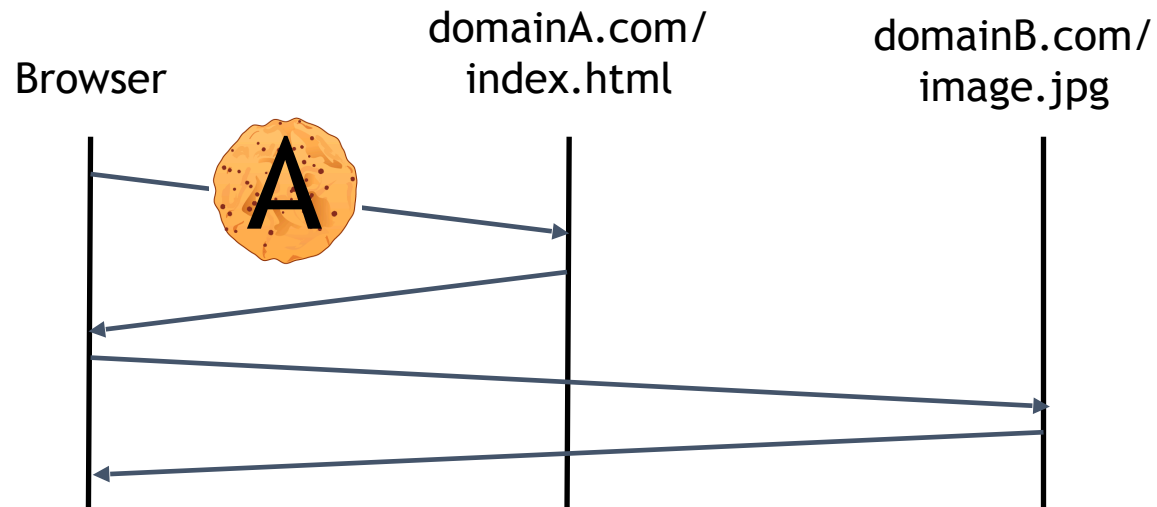


Web Fundamentals

HTTP cookies

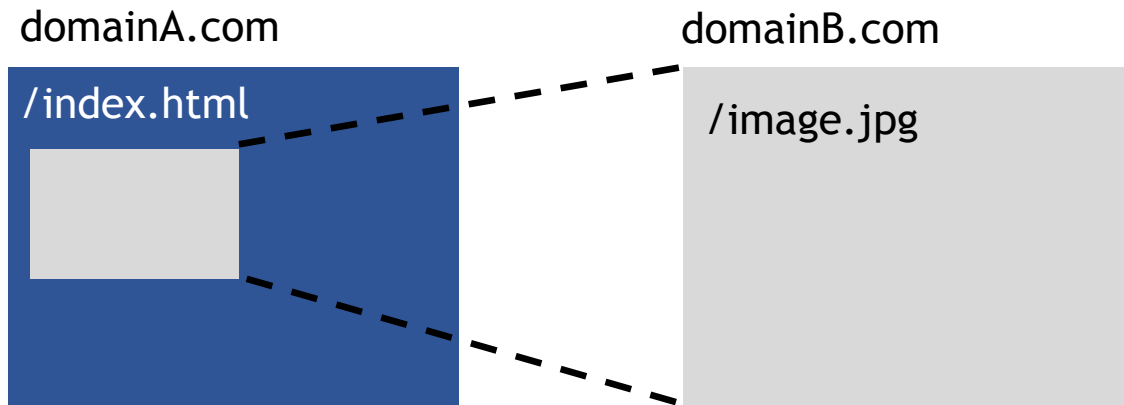


- › Implicit inclusion
- › Authentication / identification
- › Same-Origin Policy

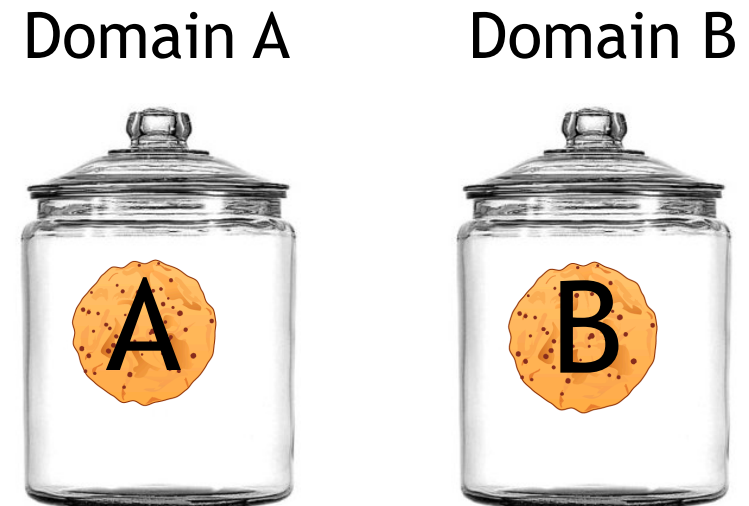
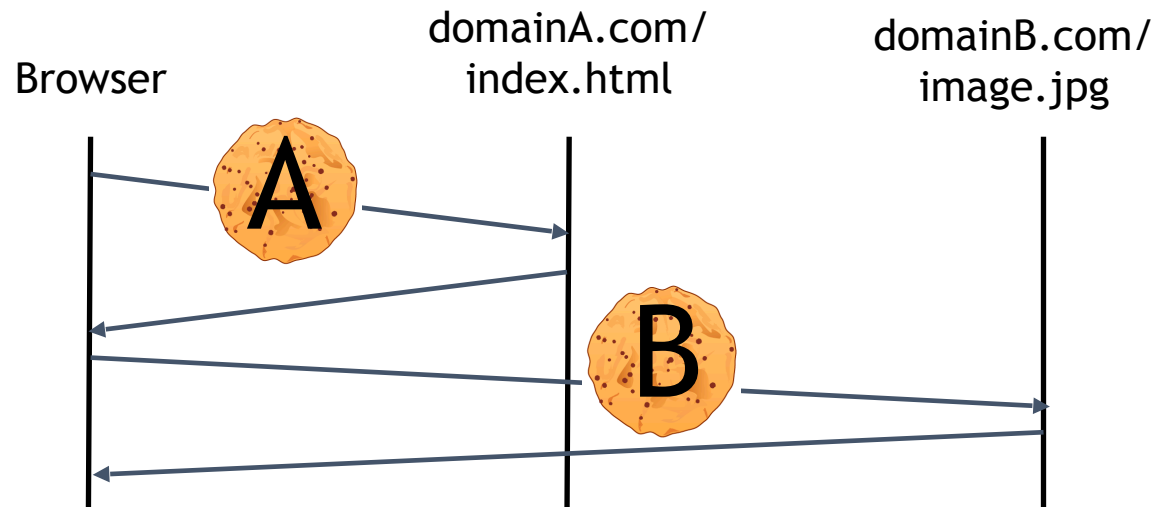


Web Fundamentals

HTTP cookies



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



Cross-site request forgery (CSRF)

Consequence

Attacker can perform authenticated actions with the victim's account

Context

Vulnerabilities detected for big companies in the past (YouTube, banking sites, Netflix, ...)

Cross-site request forgery (CSRF)

Consequence

Attacker can perform authenticated actions with the victim's account

Context

Vulnerabilities detected for big companies in the past (YouTube, banking sites, Netflix, ...)



victim

cute-kittens.com

Cross-site request forgery (CSRF)

Consequence

Attacker can perform authenticated actions with the victim's account

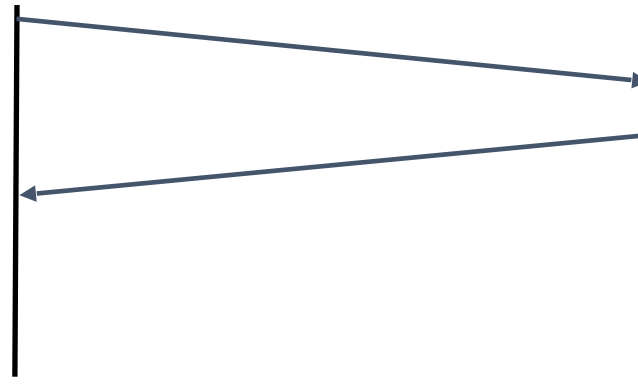
Context

Vulnerabilities detected for big companies in the past (YouTube, banking sites, Netflix, ...)



victim

cute-kittens.com



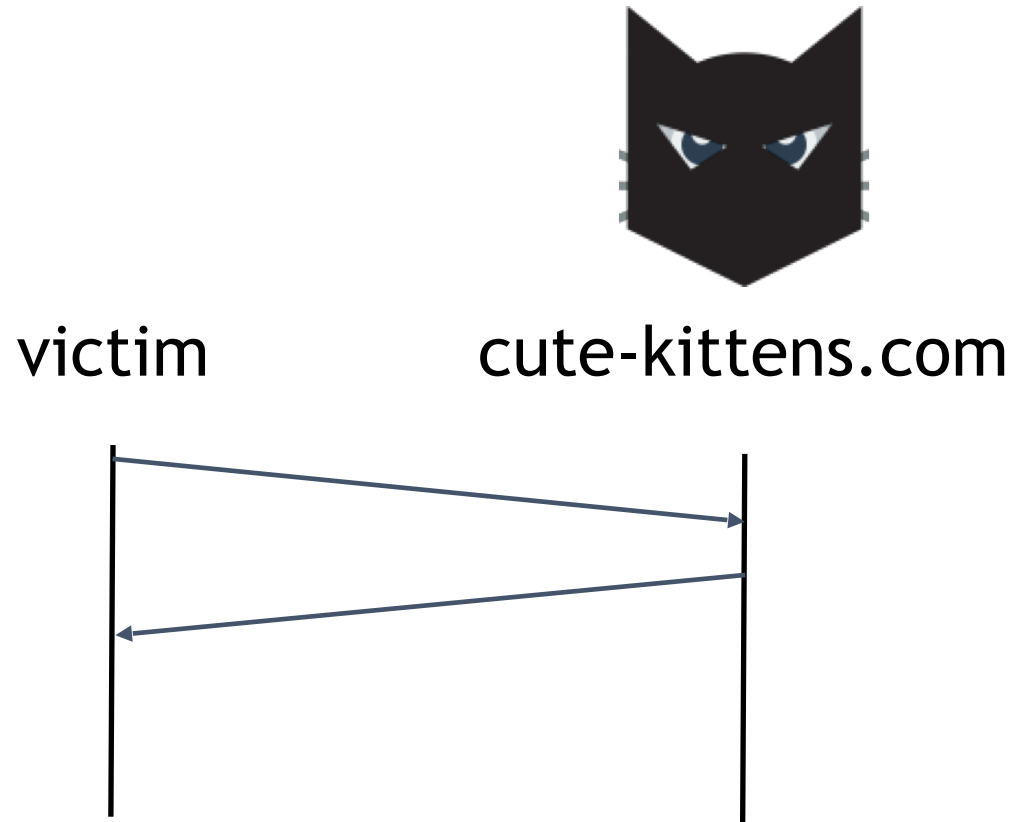
Cross-site request forgery (CSRF)

Consequence

Attacker can perform authenticated actions with the victim's account

Context

Vulnerabilities detected for big companies in the past (YouTube, banking sites, Netflix, ...)



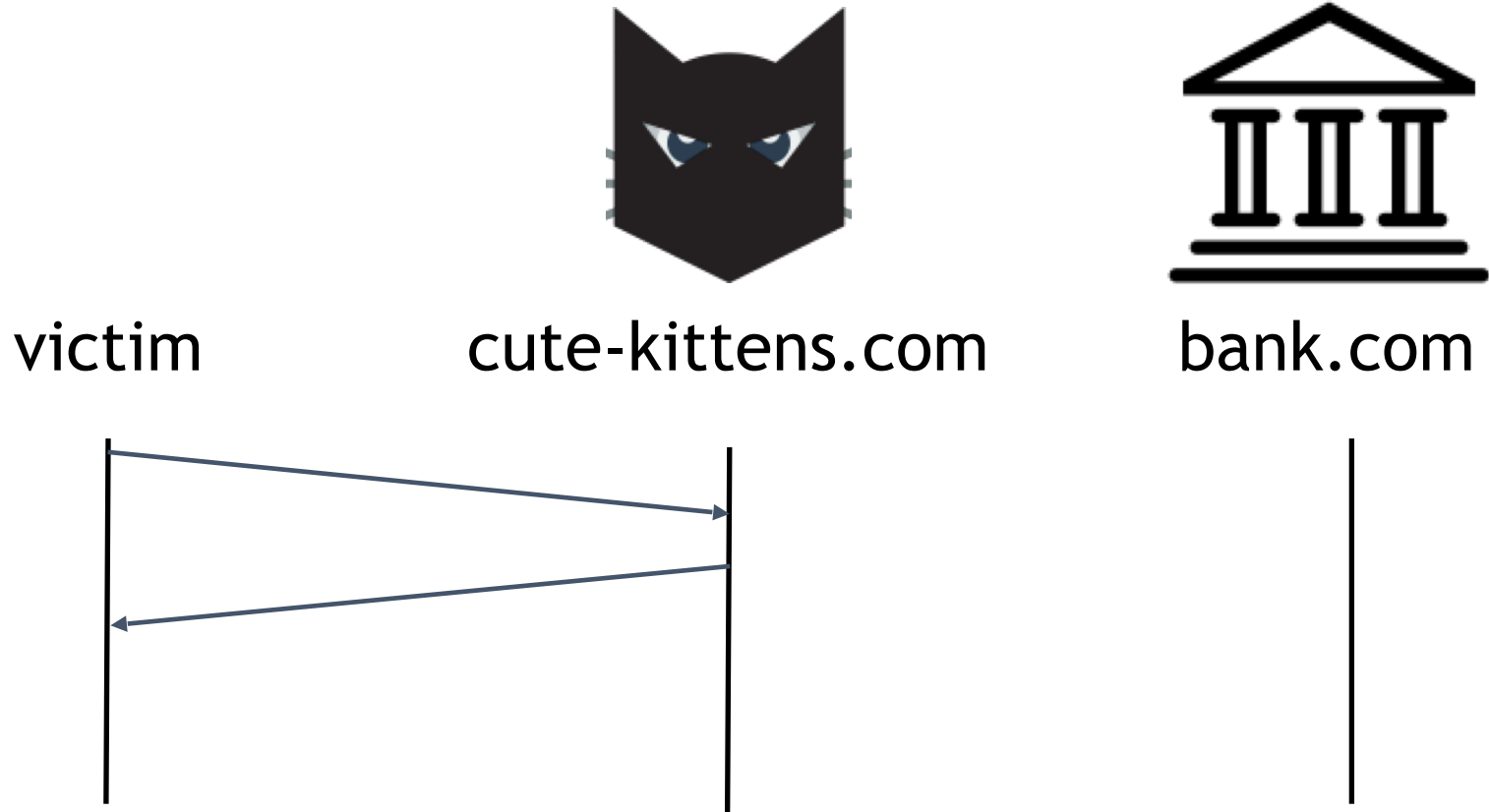
Cross-site request forgery (CSRF)

Consequence

Attacker can perform authenticated actions with the victim's account

Context

Vulnerabilities detected for big companies in the past (YouTube, banking sites, Netflix, ...)



```

```

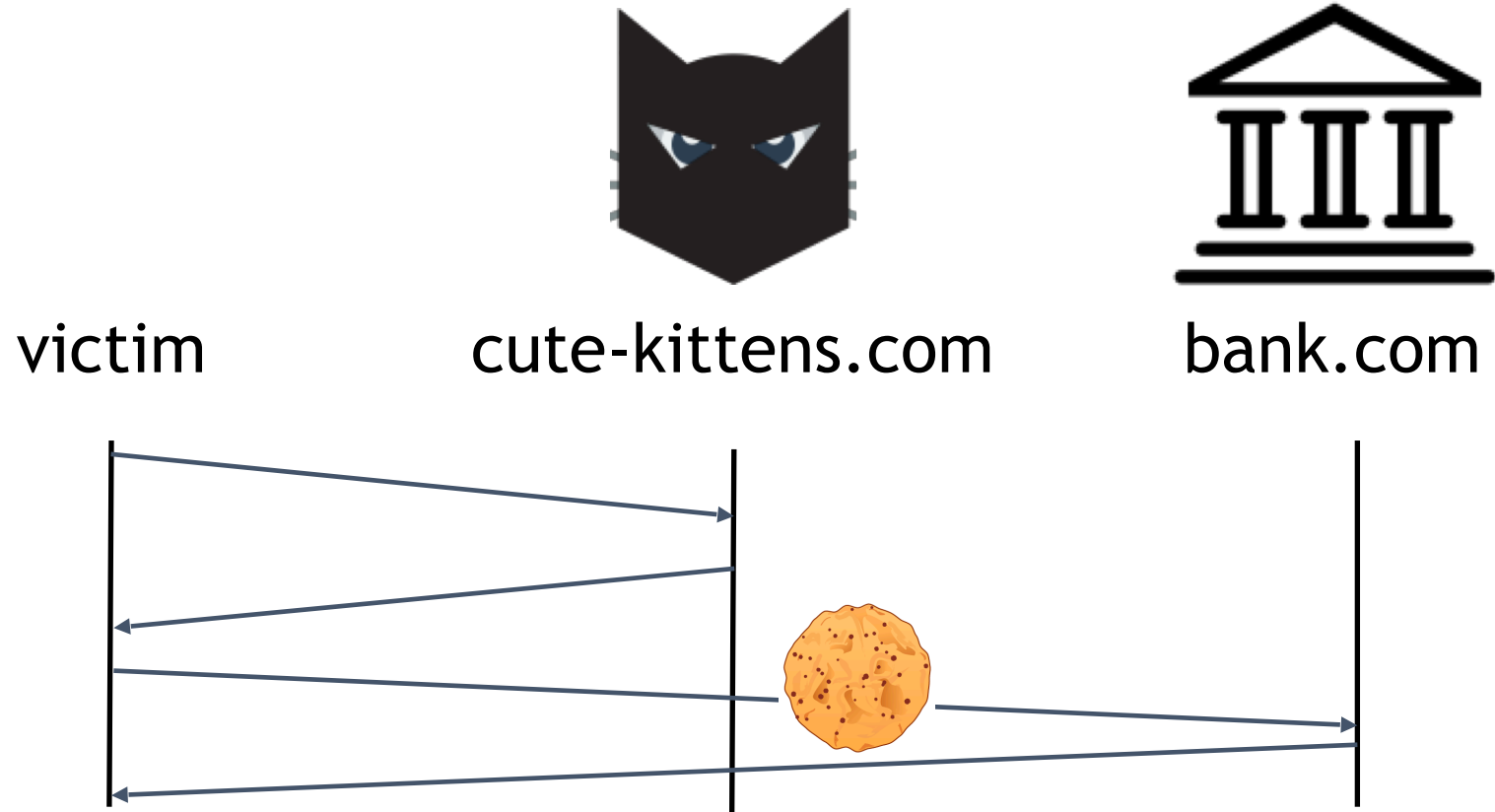
Cross-site request forgery (CSRF)

Consequence

Attacker can perform authenticated actions with the victim's account

Context

Vulnerabilities detected for big companies in the past (YouTube, banking sites, Netflix, ...)



```

```


Same-site cookie

- › Cookie with additional attribute: SameSite
- › Instructed server-side, enforced client-side
 - ›› SameSite=strict → cookie not included in any cross-site requests
 - ›› SameSite=lax → exceptions: top-level GET, prerender
- › Google Chrome intends to make SameSite=lax the default
 - ›› Starting from version 76

Use of same-site cookies

bank.com



victim



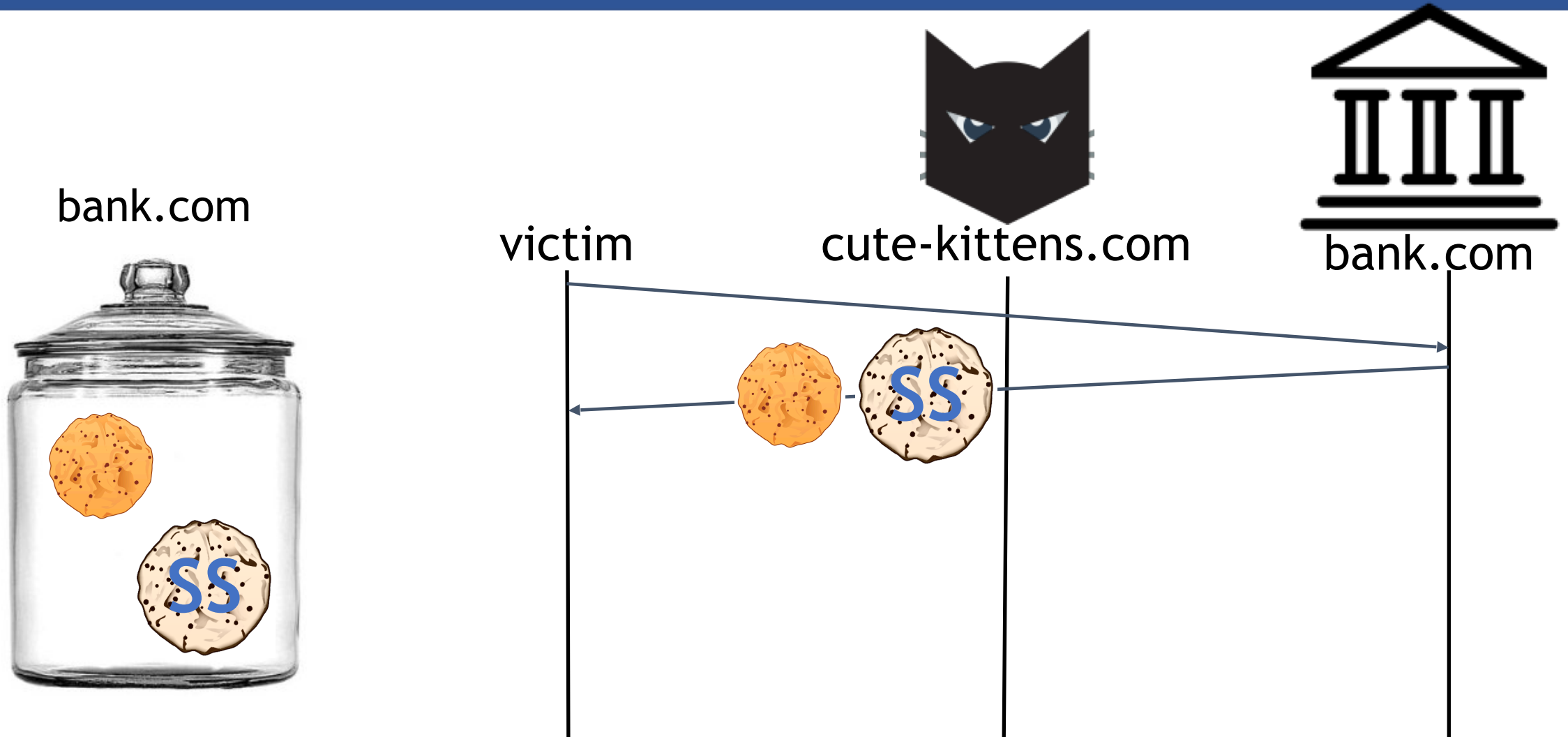
cute-kittens.com



bank.com

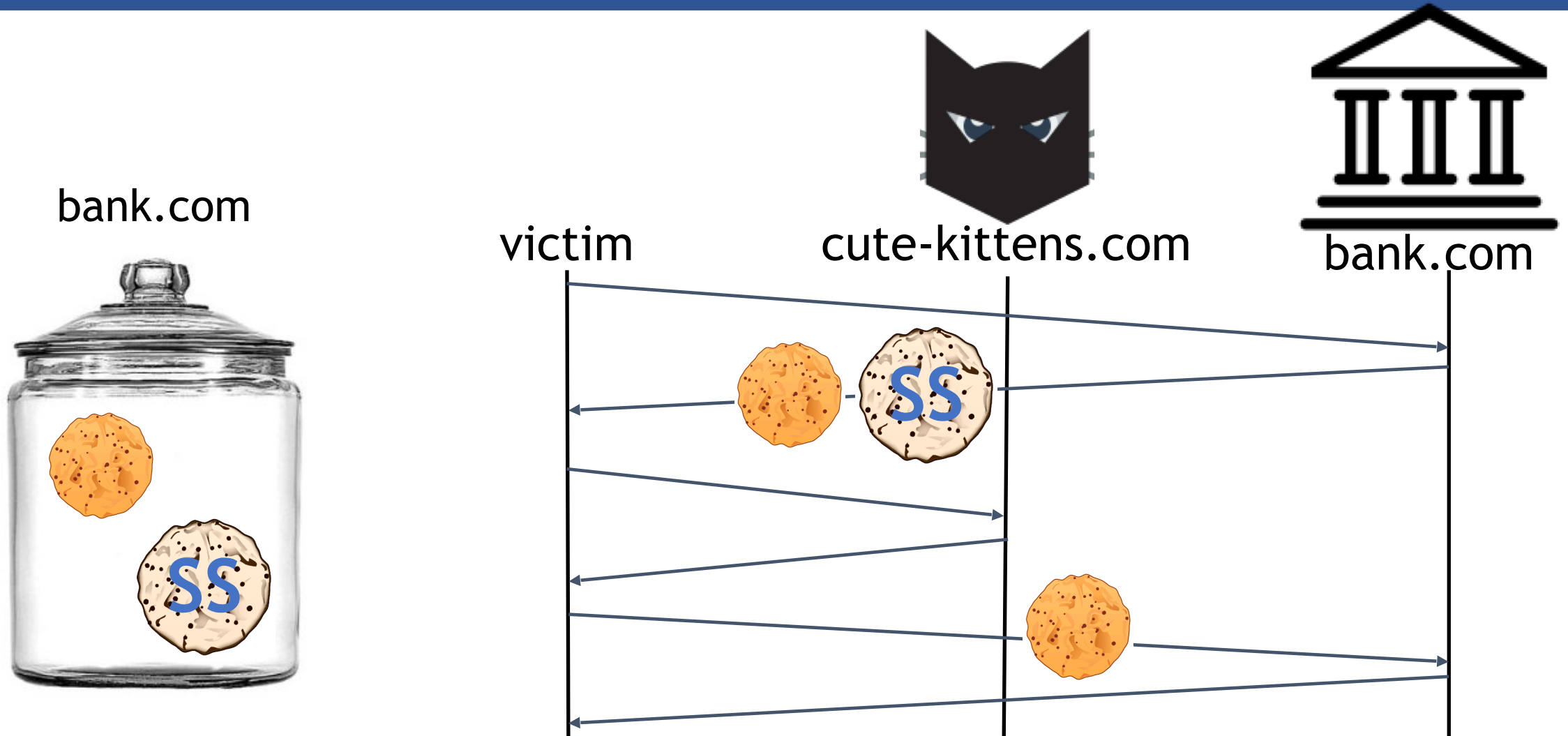


Use of same-site cookies



Set-Cookie: auth=ekSd2lksq090pQDs; SameSite=lax

Use of same-site cookies



Set-Cookie: auth=ekSd2lksq090pQDs; SameSite=lax

Why evaluate third-party cookie policies?

- › Browsers are known to exhibit inconsistent behavior
 - ›› Deviate from standards
 - ›› Unintended side-effects of certain features
- › Extensions have been actively bypassed in the past
 - ›› Pornhub exploited WebSockets to circumvent adblockers [1]

Why evaluate third-party cookie policies?

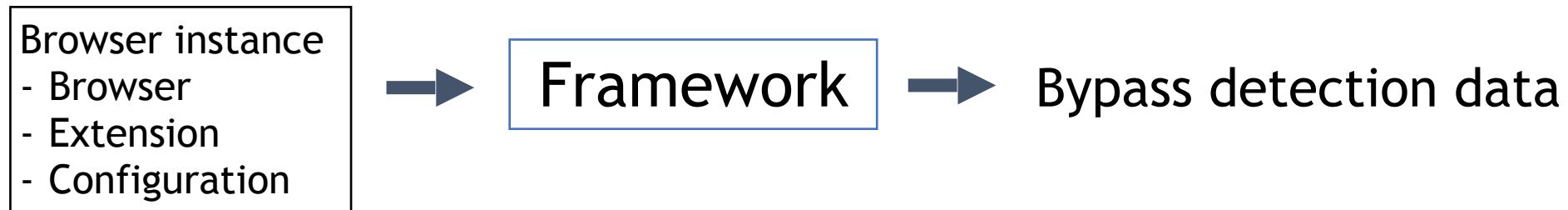
- › Browsers are known to exhibit inconsistent behavior
 - ›› Deviate from standards
 - ›› Unintended side-effects of certain features
- › Extensions have been actively bypassed in the past
 - ›› Pornhub exploited WebSockets to circumvent adblockers [1]

=> Comprehensive evaluation of effectiveness needed!

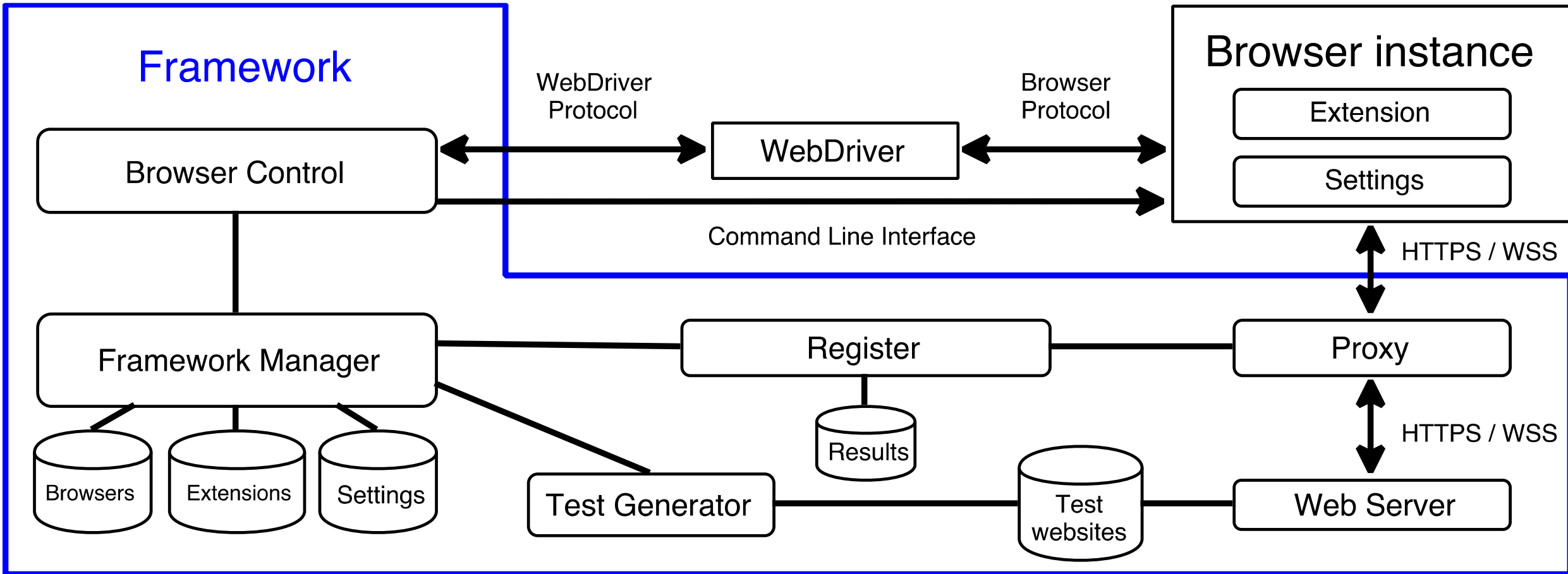
Cookie policy testing framework

Framework requirements

- › Black box
 - ›› Not all browser source code is available
 - ›› Browsers consist of millions of lines of code
- › Needs to support browser extensions



Framework design










Test-case generation

- › Initiate requests using different mechanisms
- › Application Cache
 - ›› Allows cross-origin caching
- › HTML tags
 - ›› <script>, , <link>, ...
- › Headers
 - ›› Link, CSP report, ...
- › Redirects
- › JavaScript
 - ›› Fetch(), EventSource, ...
- › PDF JS
 - ›› sendForm()
- › ServiceWorker API

Tested browser instances

> Browsers

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

> Extensions

>> Ad blocking



>> Tracking protection



Results

Results: built-in browser policies

- › Blocking third-party cookies
 - ›› bypassed in Chrome & Opera by JS in PDF (`sendForm()`)
 - ›› Safari 10 & Edge 40: completely unfunctional
- › Built-in tracking protection/ad blocking
 - ›› Opera Ad Blocker & Firefox Tracking Protection: bypasses in several categories
 - ››› E.g. Link: `<http://tracker.com/track/>; rel="prev"`

Results: browser extensions

- › All extensions could be bypassed
- › Design flaws
 - ›› In Chrome, PDFs are rendered in extension + impossible to intercept requests by other requests
 - ›› In other browsers: certain requests bypass the extensions
- › Unclear API
 - ›› No clear distinction for browser background requests
- › Common mistakes
 - ›› Insufficient permissions to intercept certain requests

PDFium design flaw



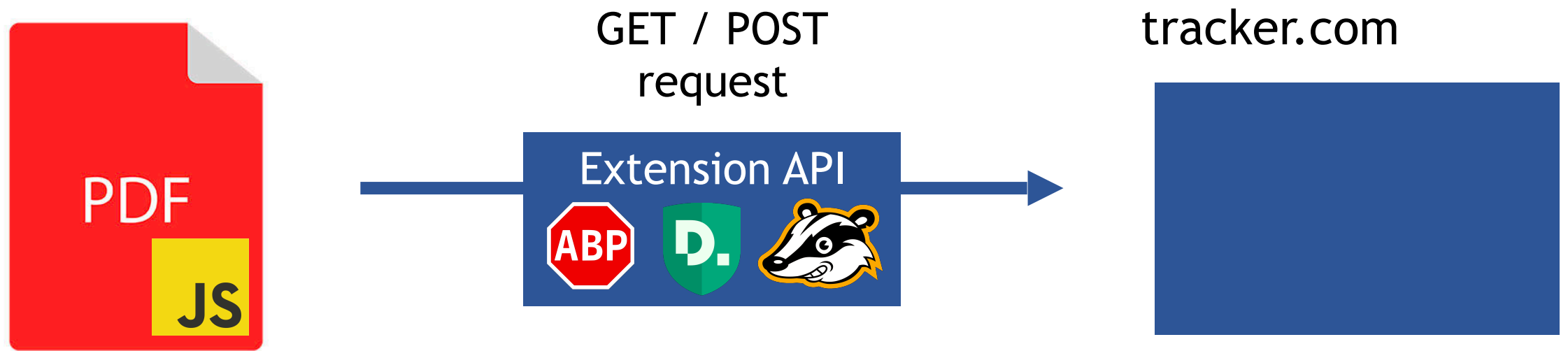
GET / POST
request



tracker.com



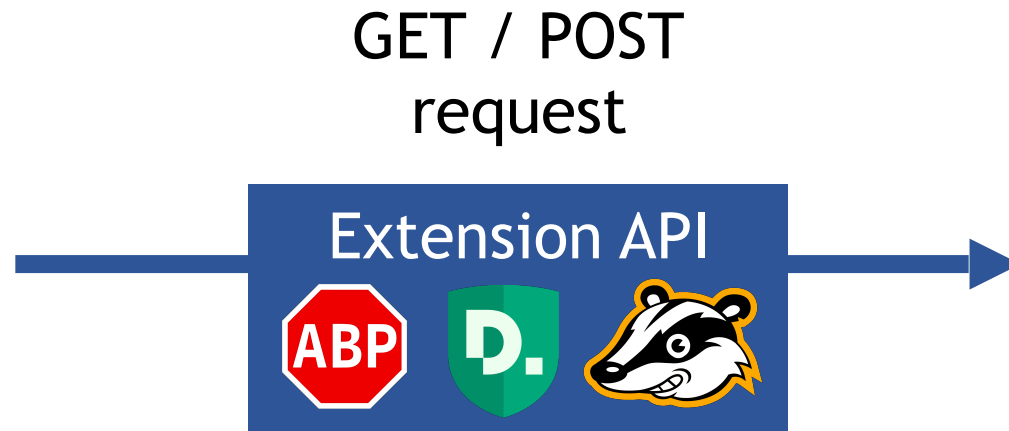
PDFium design flaw



PDFium design flaw



Plugin / Extension



tracker.com



PDFium design flaw



Plugin / Extension



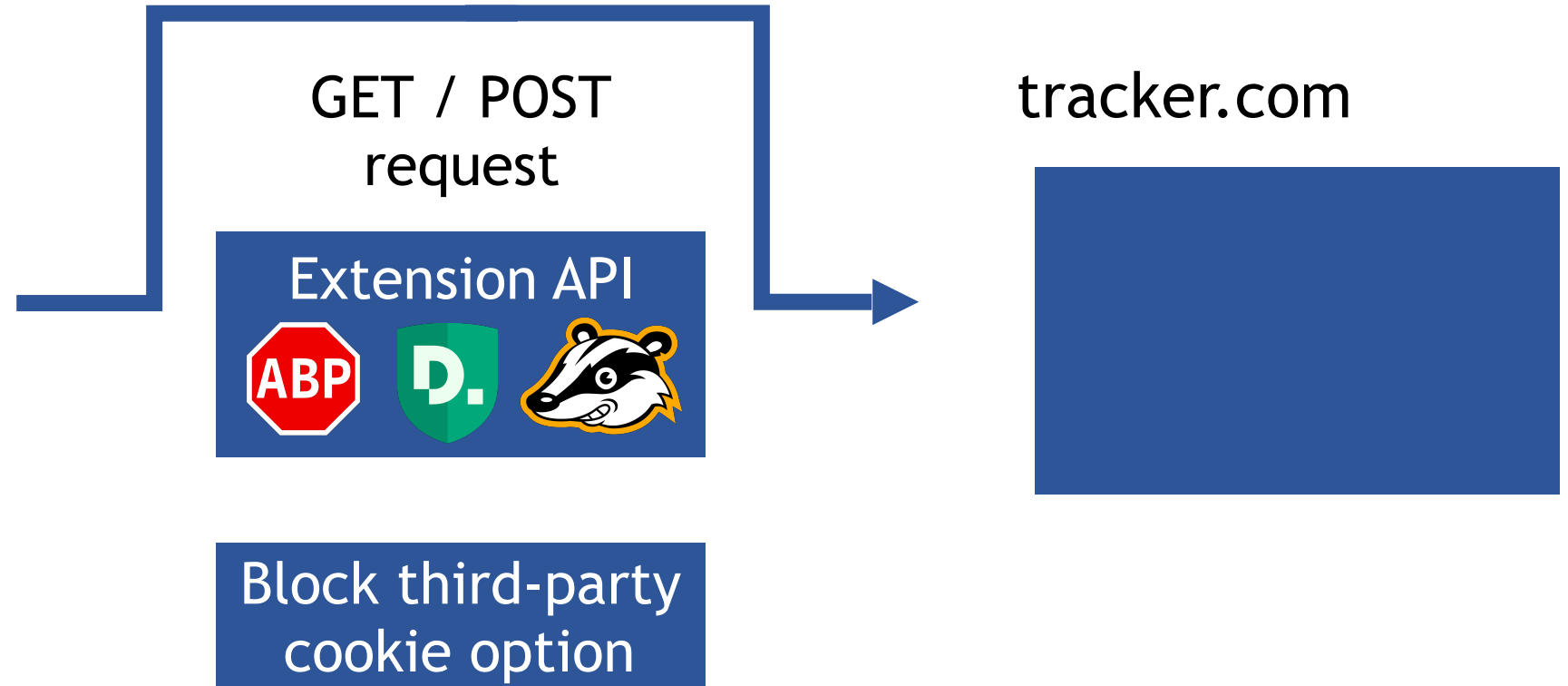
tracker.com



PDFium design flaw

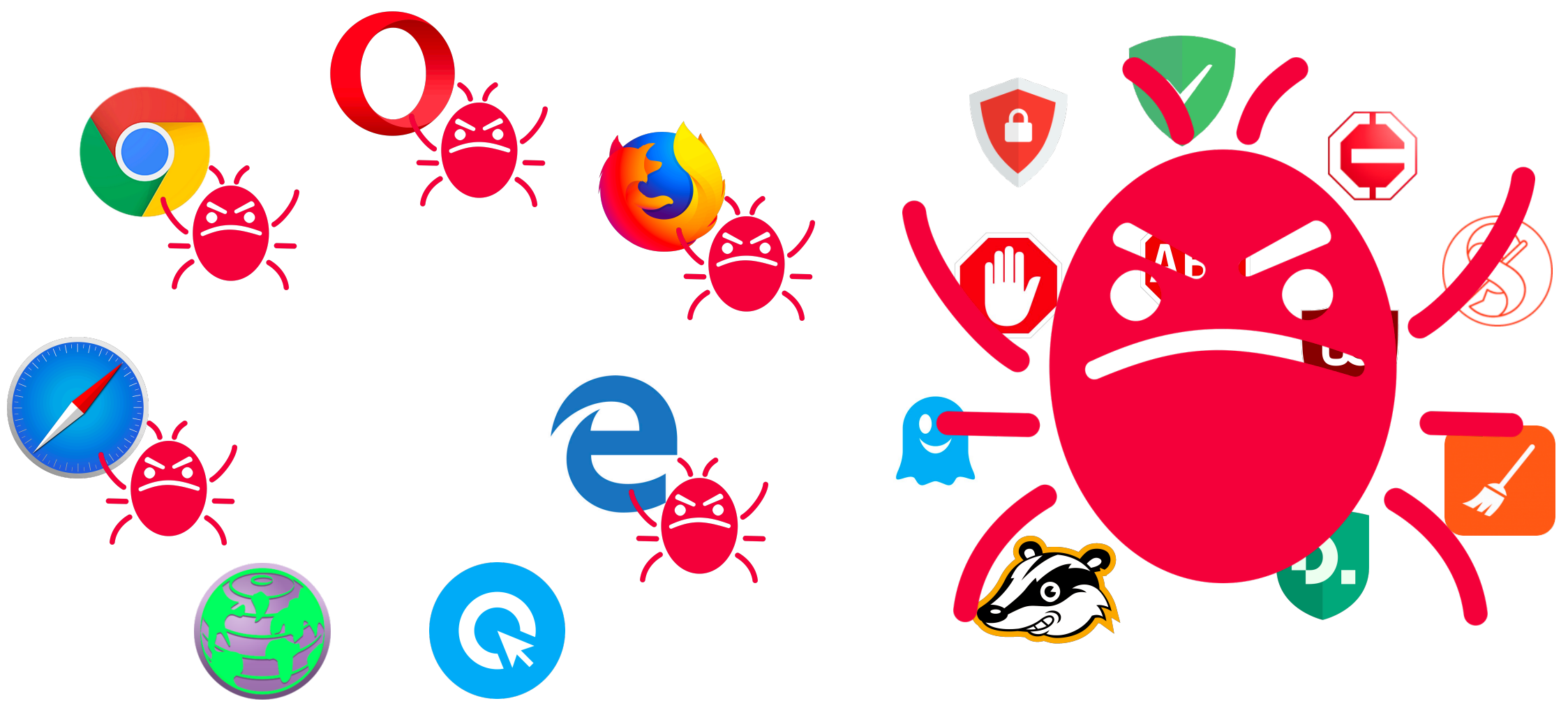


Plugin / Extension



Results: same-site cookie

- › Chrome & Opera
 - ›› SameSite=strict cookie is sent for prerender requests
- › Edge
 - ›› SameSite=lax bypasses: WebSocket API, <embed>, <object>
 - ›› SameSite=strict bypasses: WebSocket API, redirects

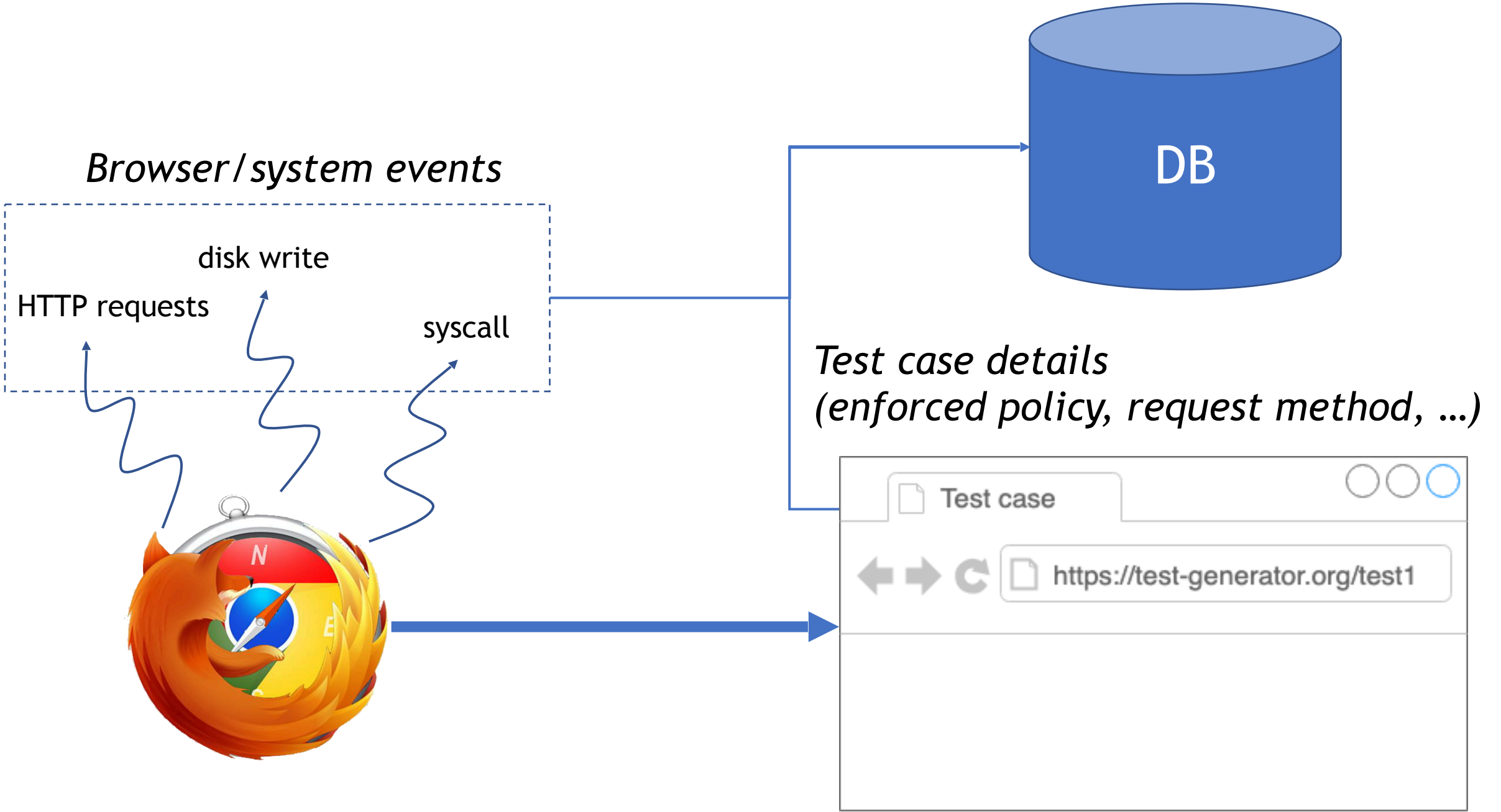


Check out our paper:
jar_usenix18.pdf

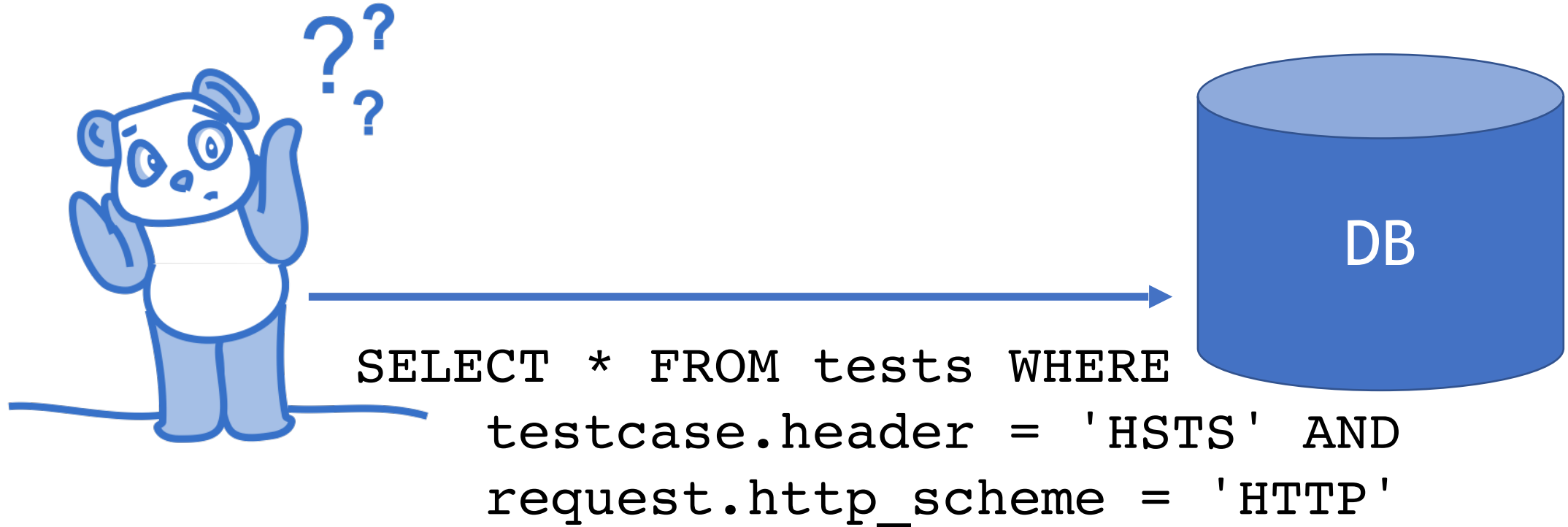
https://tom.vg/papers/who-left-open-the-cookie-jar_usenix18.pdf

short URL: <https://bit.ly/owasp-cookie-paper>

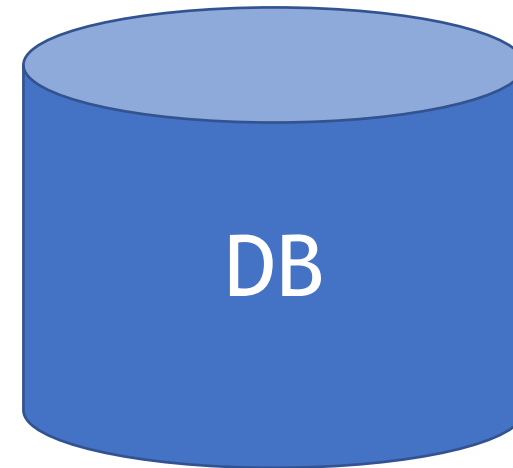
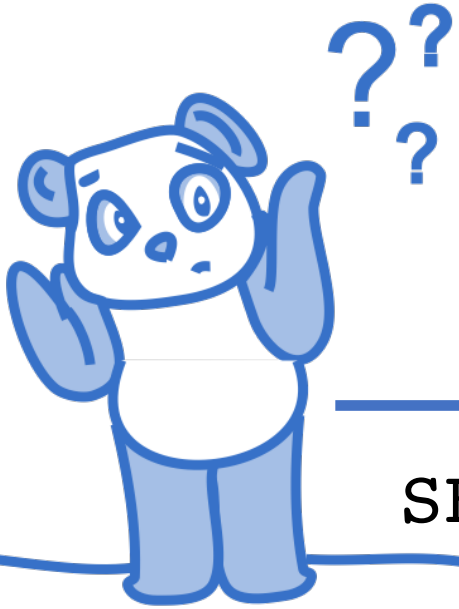
Evaluating other browser
security/privacy policies



Is Strict-Transport-Security correctly implemented in all browsers?



Is CSP's `img-src` directive correctly implemented in all browsers



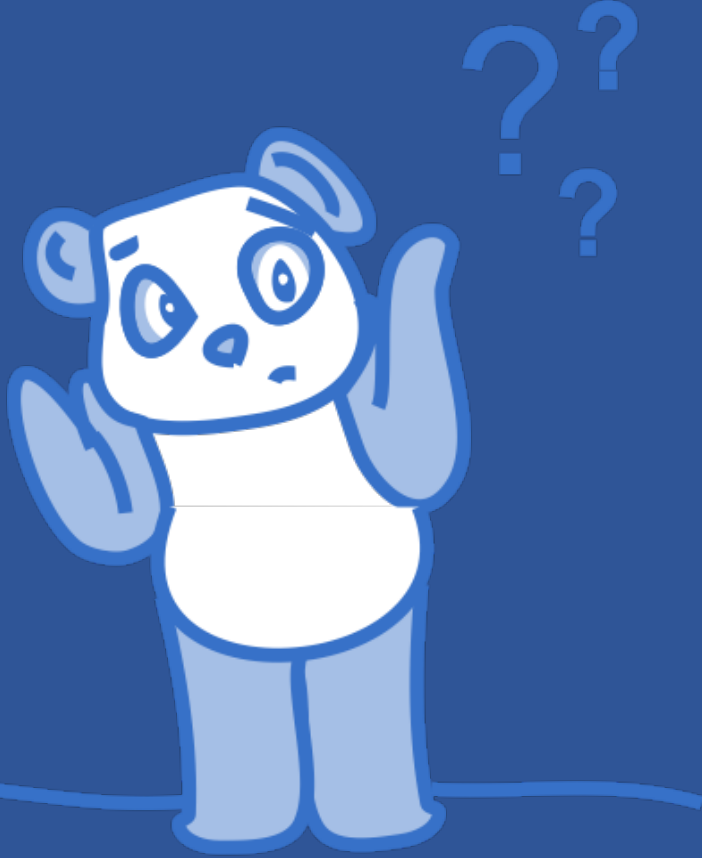
```
SELECT * FROM tests WHERE  
testcase.header = 'CSP' AND  
testcase.value CONTAINS 'img-src: none' AND  
request.type = 'image'
```

Browser evaluation framework

- › Validate correctness of enforcement of implicit/explicit policies
- › Supports all browsers
 - ›› Various configurations
 - ›› Measure influence of browser extensions
- › Request triggers can be fuzzed for completeness
- › Can be used to validate browser implementation before release
 - ›› New features may introduce side-effects in policies (e.g. prerender)

Conclusion

- › Browsers are very complex
 - ›› Many APIs/features, millions LoC
- › Extensive evaluation is required
 - ›› Should cover entire “ecosystem”: different request mechanisms, browser extensions, ...
- › Several issues discovered for cookie policies
 - ›› Bypasses for all browser extensions + several built-in browser policies
- › Framework for evaluating browser security/privacy policies



Questions?

<https://WhoLeftOpenTheCookieJar.eu>



@tomvangoethem
@GJFR_

DistriNet