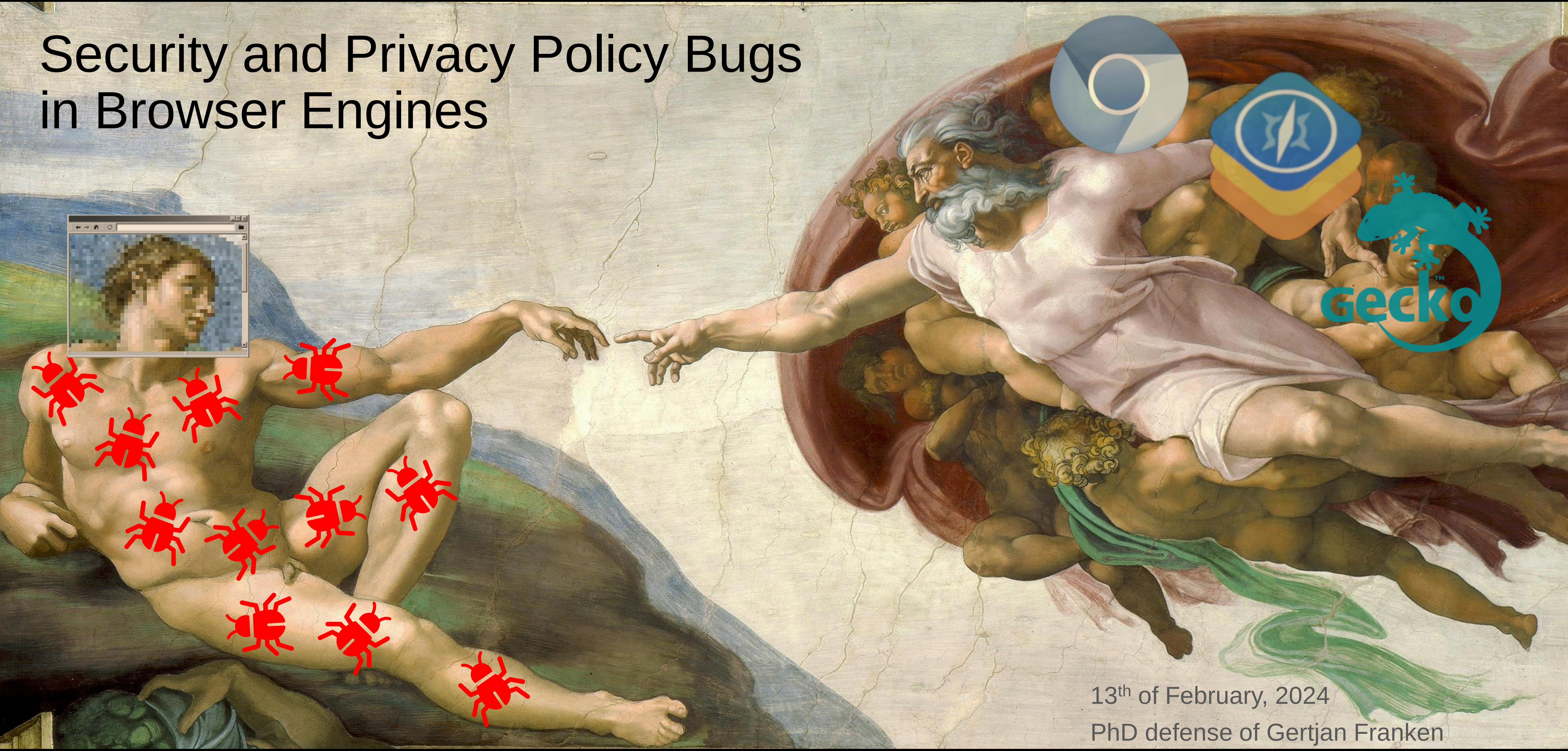
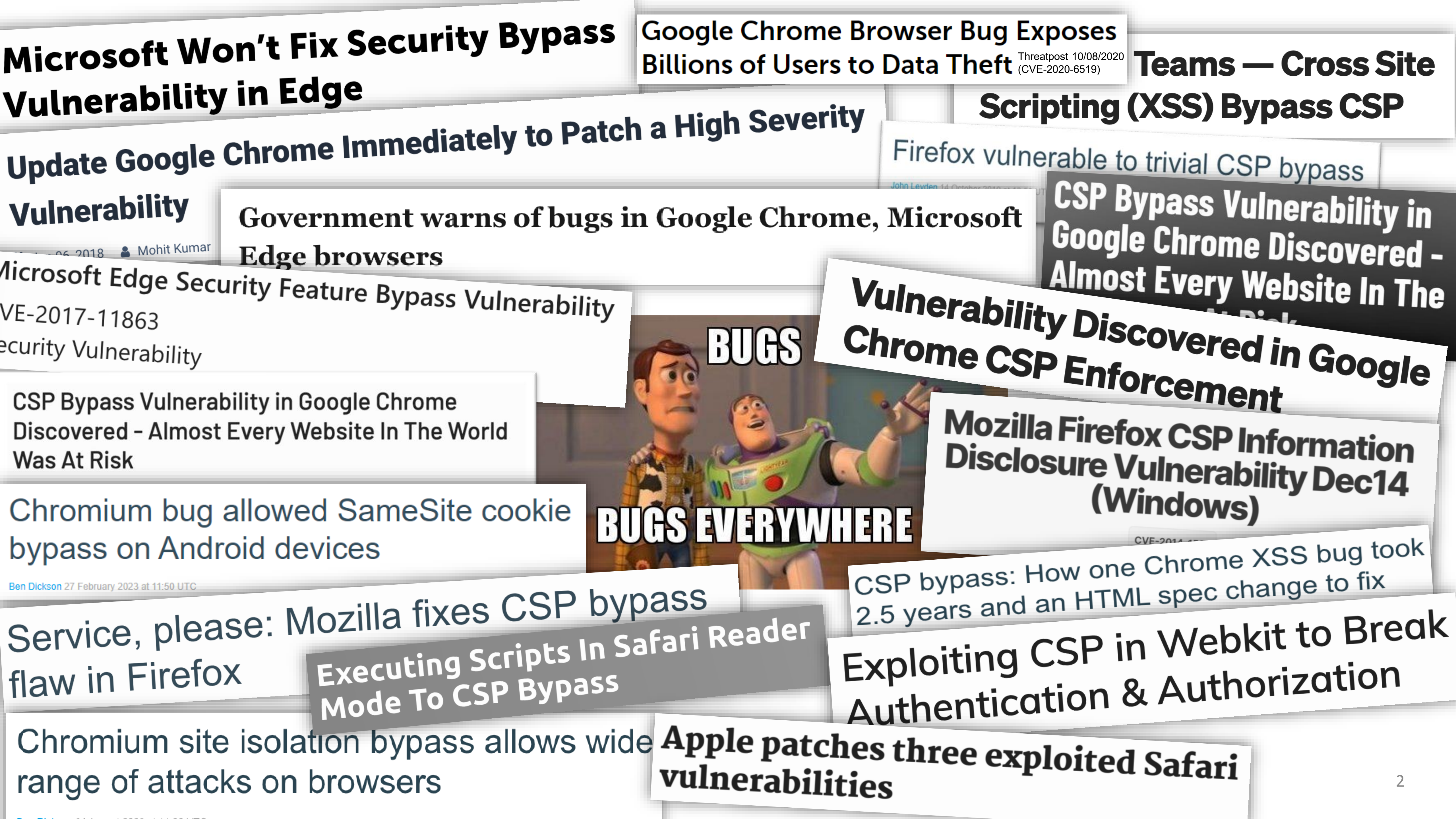
Security and Privacy Policy Bugs in Browser Engines



13th of February, 2024

PhD defense of Gertjan Franken



Microsoft Won't Fix Security Bypass Vulnerability in Edge

Google Chrome Browser Bug Exposes Billions of Users to Data Theft Threatpost 10/08/2020 (CVE-2020-6519)

Teams — Cross Site Scripting (XSS) Bypass CSP

Update Google Chrome Immediately to Patch a High Severity Vulnerability

06, 2018 Mohit Kumar

Firefox vulnerable to trivial CSP bypass

John Leyden 14 October 2019

Government warns of bugs in Google Chrome, Microsoft Edge browsers

CSP Bypass Vulnerability in Google Chrome Discovered - Almost Every Website In The

icrosoft Edge Security Feature Bypass Vulnerability

VE-2017-11863

ecurity Vulnerability

Vulnerability Discovered in Google Chrome CSP Enforcement

CSP Bypass Vulnerability in Google Chrome Discovered - Almost Every Website In The World Was At Risk

Mozilla Firefox CSP Information Disclosure Vulnerability Dec14 (Windows)

CVE-2014

Chromium bug allowed SameSite cookie bypass on Android devices

Ben Dickson 27 February 2023 at 11:50 UTC

CSP bypass: How one Chrome XSS bug took 2.5 years and an HTML spec change to fix

Service, please: Mozilla fixes CSP bypass flaw in Firefox

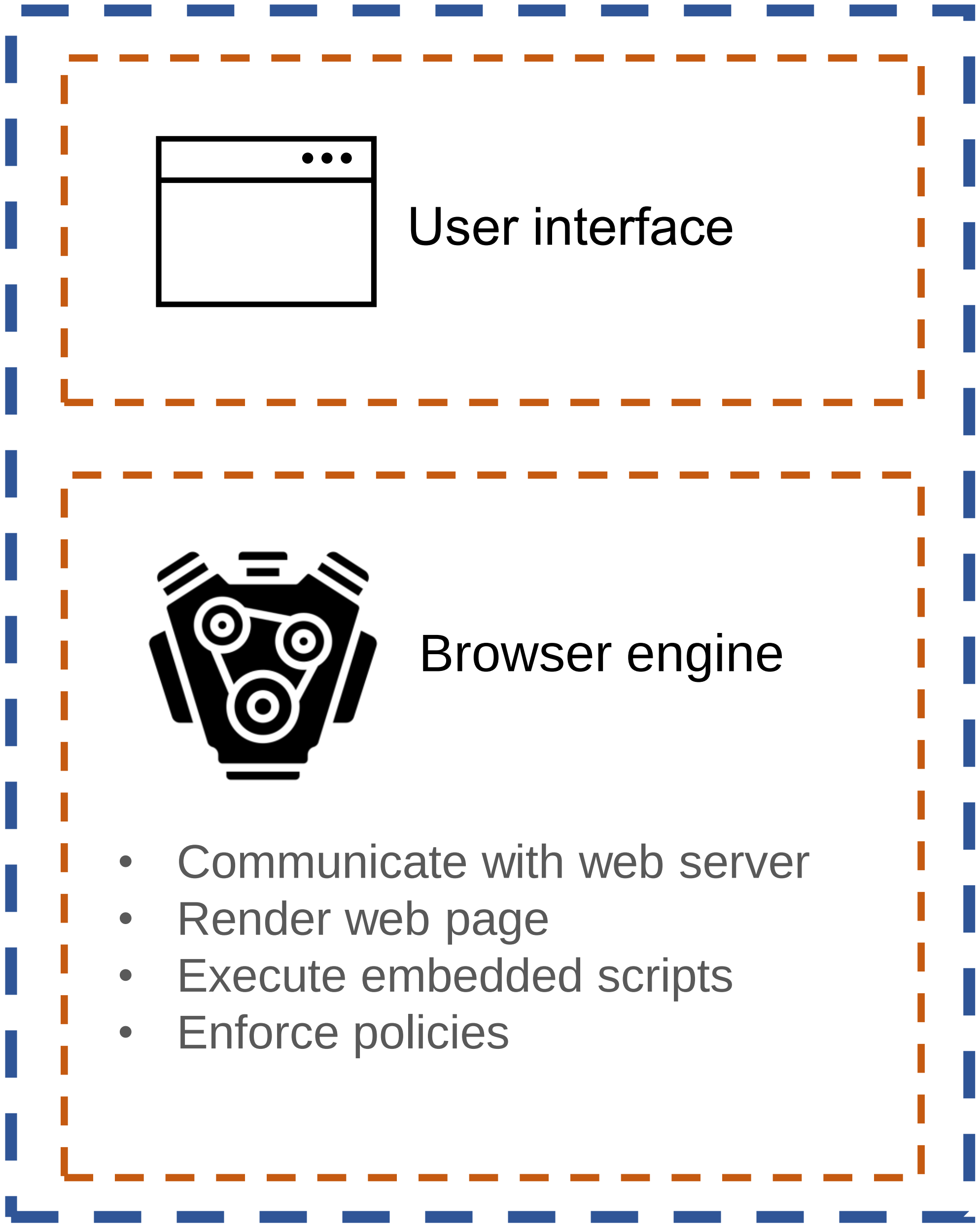
Executing Scripts In Safari Reader Mode To CSP Bypass

Exploiting CSP in Webkit to Break Authentication & Authorization

Chromium site isolation bypass allows wide range of attacks on browsers

Apple patches three exploited Safari vulnerabilities

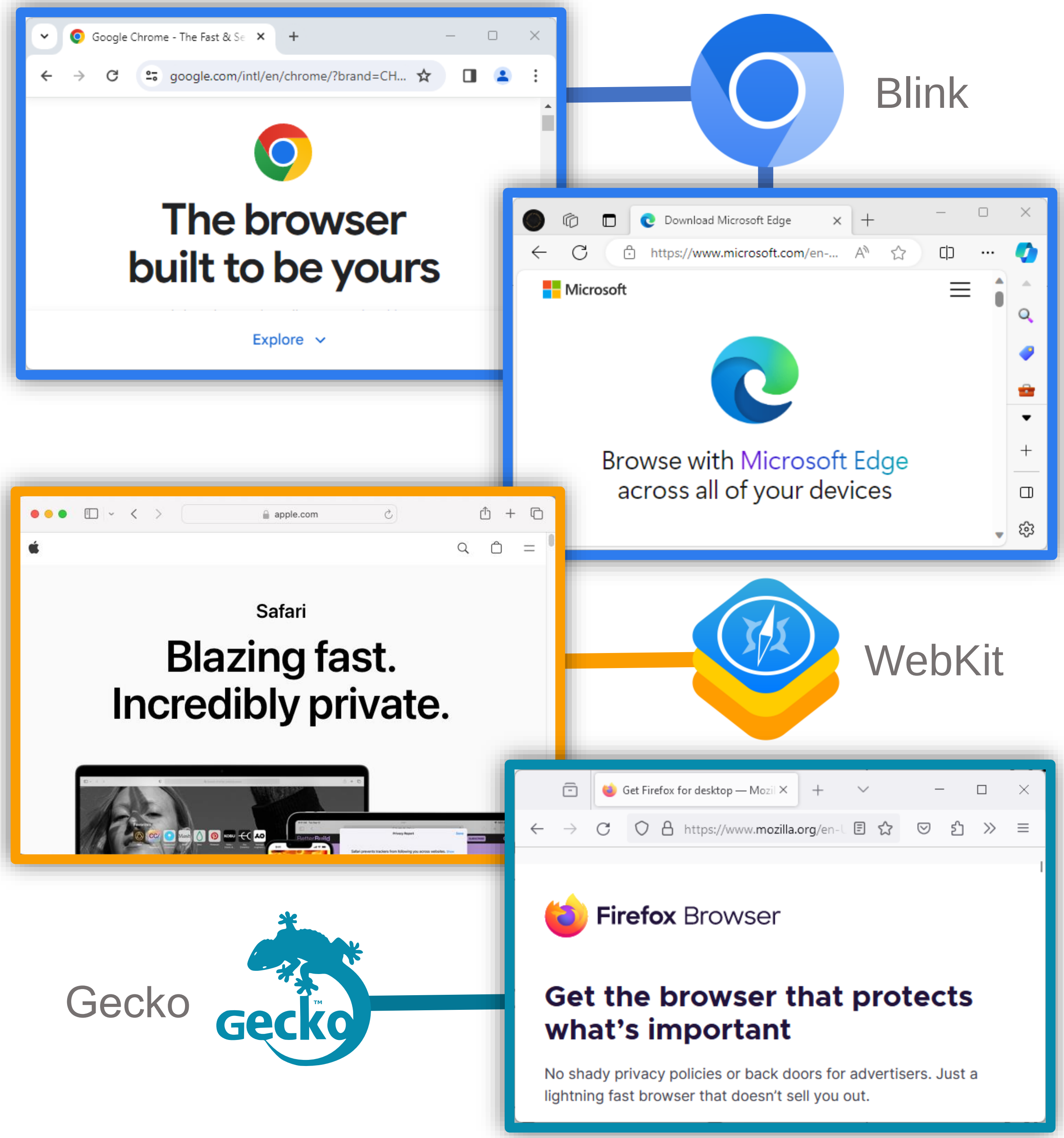
Web browser



User interface

Browser engine

- Communicate with web server
- Render web page
- Execute embedded scripts
- Enforce policies



Google Chrome - The Fast & Se
google.com/intl/en/chrome/?brand=CH...
Explore
Blink
Download Microsoft Edge
https://www.microsoft.com/en-...
Microsoft
apple.com
WebKit
Get Firefox for desktop — Mozi
https://www.mozilla.org/en-
Gecko
gecko

The browser built to be yours

Browse with Microsoft Edge across all of your devices

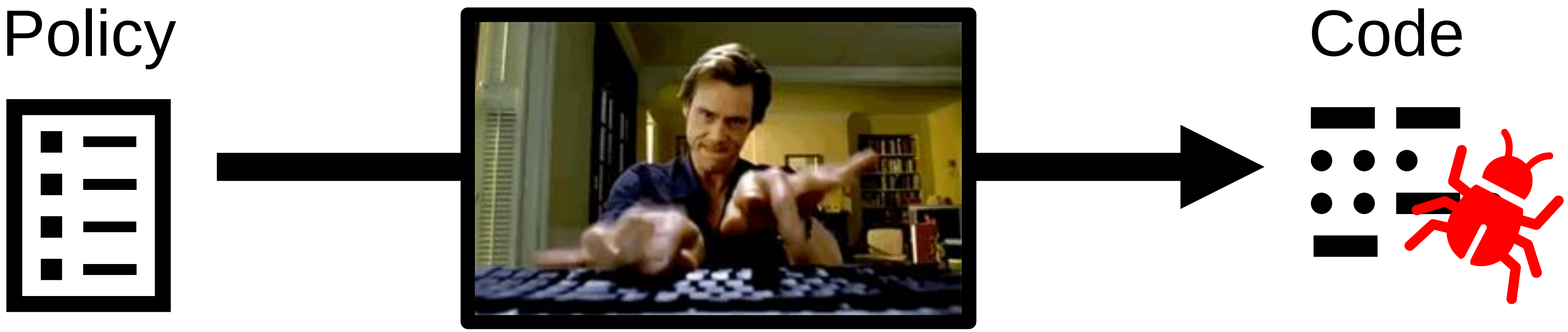
Safari

Blazing fast. Incredibly private.

Firefox Browser

Get the browser that protects what's important

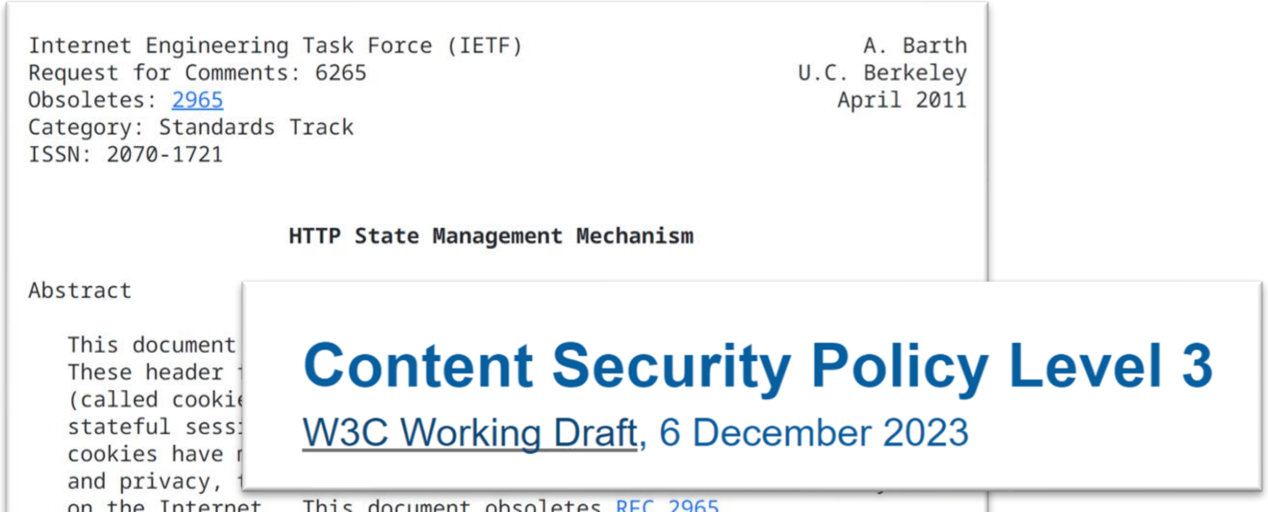
No shady privacy policies or back doors for advertisers. Just a lightning fast browser that doesn't sell you out.



Policy

Code

Web Standards



```
Internet Engineering Task Force (IETF)                          A. Barth
Request for Comments: 6265                                 U.C. Berkeley
Obsoletes: 2965                                               April 2011
Category: Standards Track
ISSN: 2070-1721


                     HTTP State Management Mechanism

Abstract

   This document
   These header
   (called cookie
   stateful sess
   cookies have
   and privacy,
   on the Internet
```

Content Security Policy Level 3

W3C Working Draft, 6 December 2023

This document obsoletes RFC 2965

Self-defined Policies

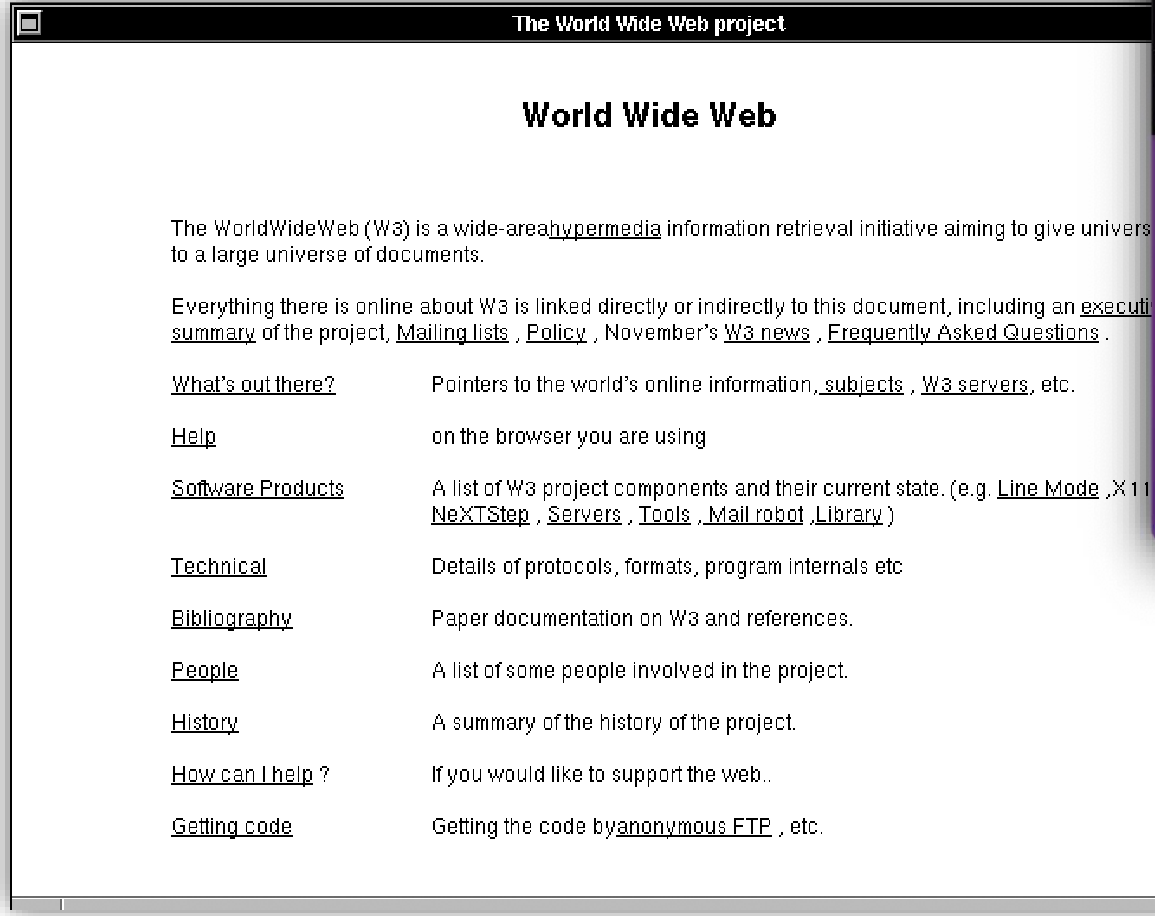
- No formal specification

Enhanced Tracking Protection

Intelligent Tracking Prevention

1991

10k lines of code



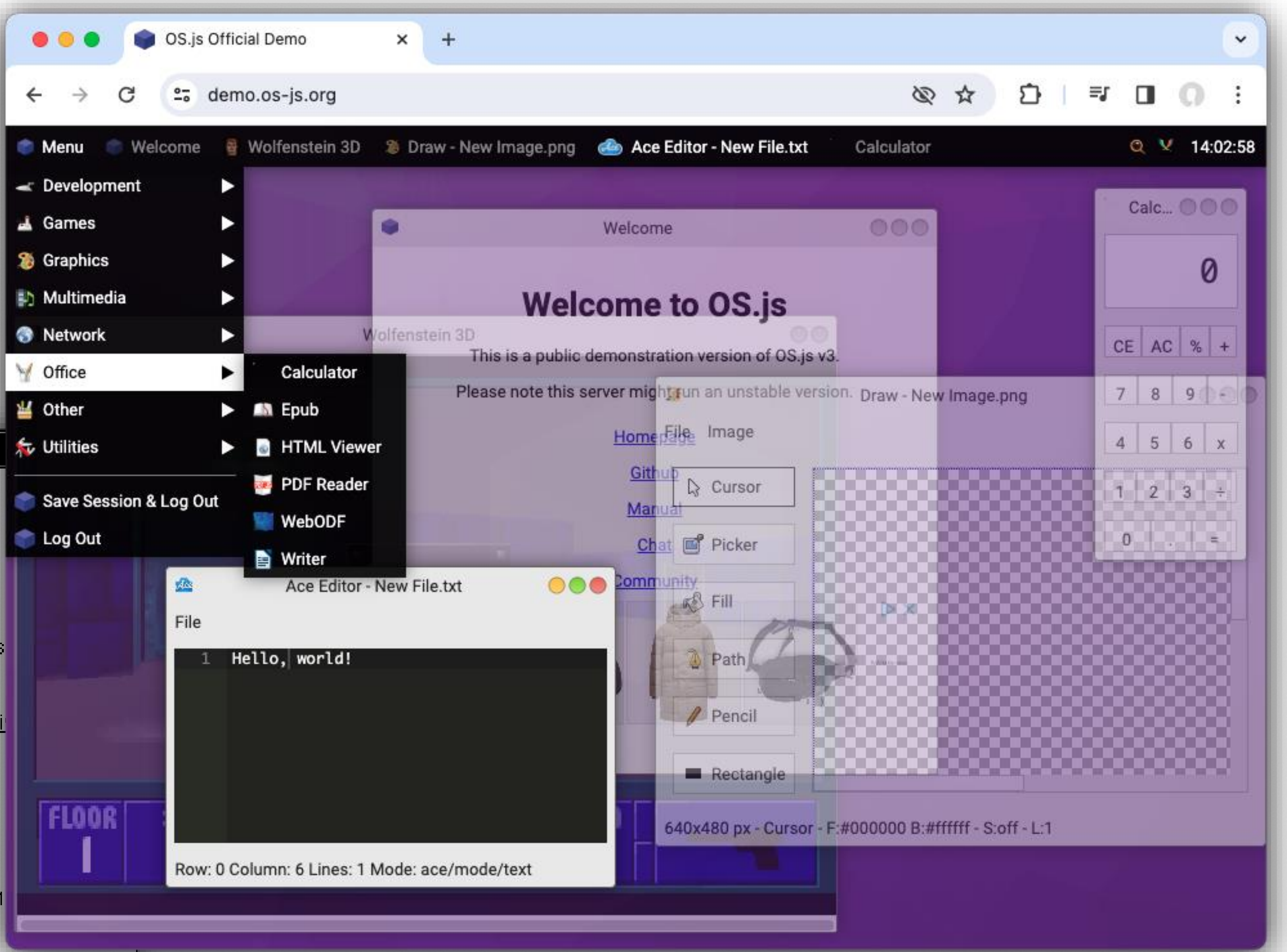
The World Wide Web project

World Wide Web

The WorldWideWeb (W3) is a wide-areahypermedia information retrieval initiative aiming to give univers to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executi summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?	Pointers to the world's online information, subjects , W3 servers, etc.
Help	on the browser you are using
Software Products	A list of W3 project components and their current state. (e.g. Line Mode ,X11 NeXTStep , Servers , Tools , Mail robot ,Library)
Technical	Details of protocols, formats, program internals etc
Bibliography	Paper documentation on W3 and references.
People	A list of some people involved in the project.
History	A summary of the history of the project.
How can I help ?	If you would like to support the web..
Getting code	Getting the code byanonymous FTP , etc.



Welcome to OS.js

This is a public demonstration version of OS.js v3.

Please note this server might run an unstable version.

Ace Editor - New File.txt

File

```
Hello, world!
```

Row: 0 Column: 6 Lines: 1 Mode: ace/mode/text

640x480 px - Cursor - F:#000000 B:#ffffff - S:off - L:1

2024

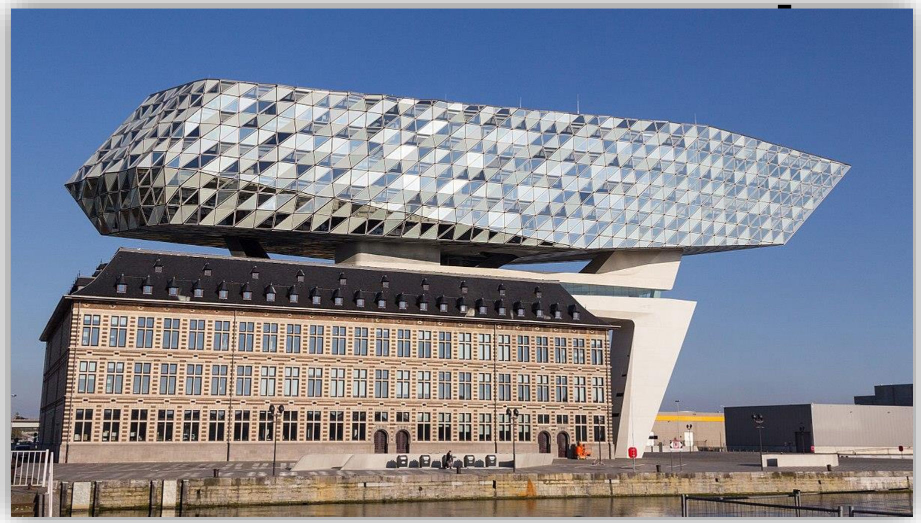
Project	Lines of code
Android	14,606k
Firefox (Gecko)	**28,049k**
Chromium (Blink)	**28,528k**
Linux kernel	34,412k

NOW

1

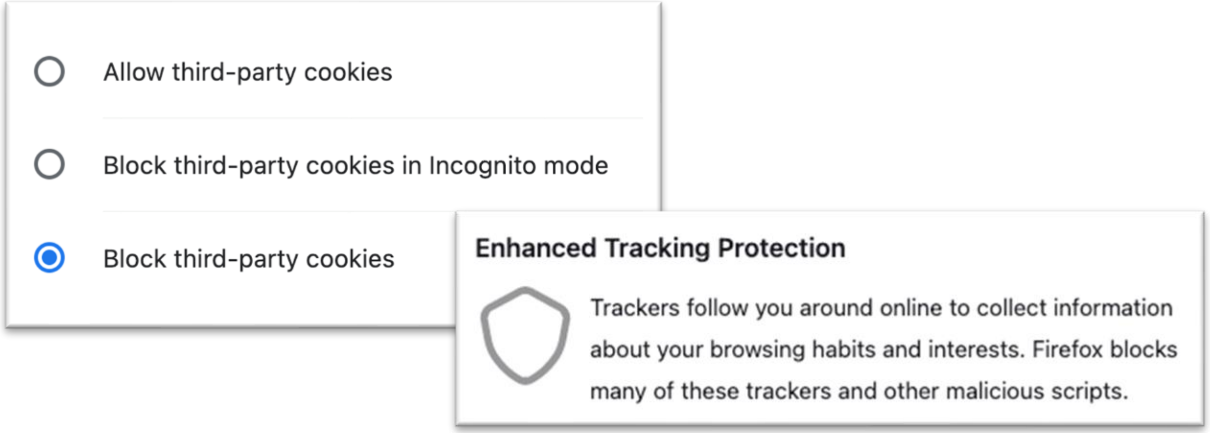
1997

2001



Evaluation of cookie and request policies

USENIX Security ‘18
[Ch2: Who Left Open the Cookie Jar?]



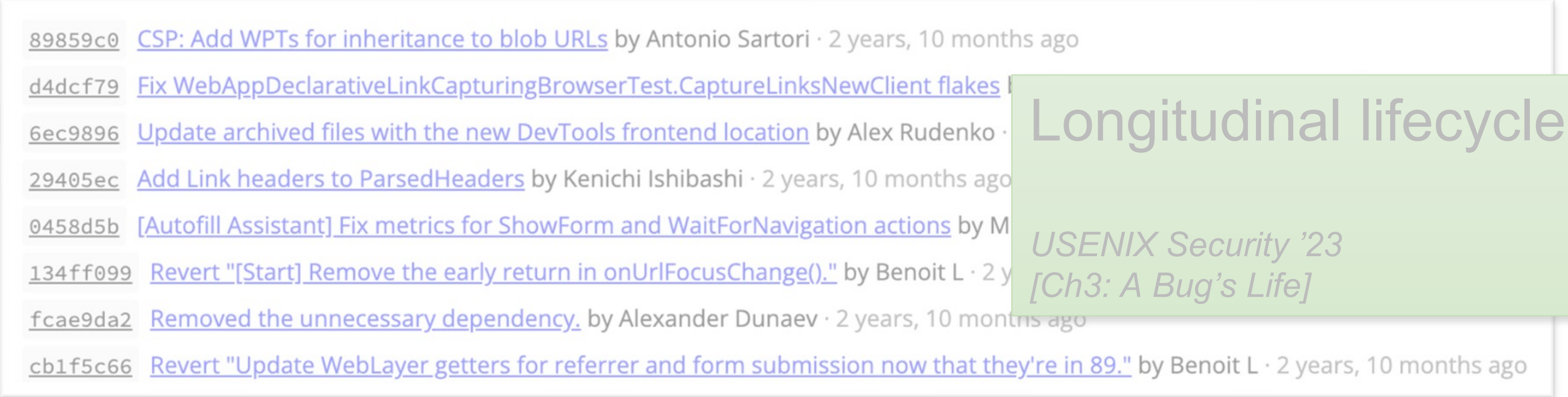
○ Allow third-party cookies

○ Block third-party cookies in Incognito mode

● Block third-party cookies

Enhanced Tracking Protection

Trackers follow you around online to collect information about your browsing habits and interests. Firefox blocks many of these trackers and other malicious scripts.



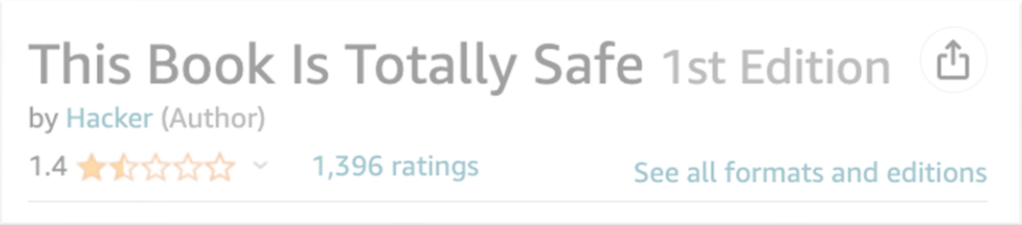
89859c0 CSP: Add WPTs for inheritance to blob URLs by Antonio Sartori · 2 years, 10 months ago
d4dcf79 Fix WebAppDeclarativeLinkCapturingBrowserTest.CaptureLinksNewClient flakes
6ec9896 Update archived files with the new DevTools frontend location by Alex Rudenko ·
29405ec Add Link headers to ParsedHeaders by Kenichi Ishibashi · 2 years, 10 months ago
0458d5b [Autofill Assistant] Fix metrics for ShowForm and WaitForNavigation actions by M
134ff099 Revert "[Start] Remove the early return in onUrlFocusChange()." by Benoit L · 2 y
fcae9da2 Removed the unnecessary dependency. by Alexander Dunaev · 2 years, 10 months ago
cb1f5c66 Revert "Update WebLayer getters for referrer and form submission now that they're in 89." by Benoit L · 2 years, 10 months ago

Longitudinal lifecycle analysis of CSP bugs

USENIX Security ’23
[Ch3: A Bug’s Life]

Security and privacy in EPUB reading systems

IEEE S&P ’21
[Ch4: Reading Between the Lines]

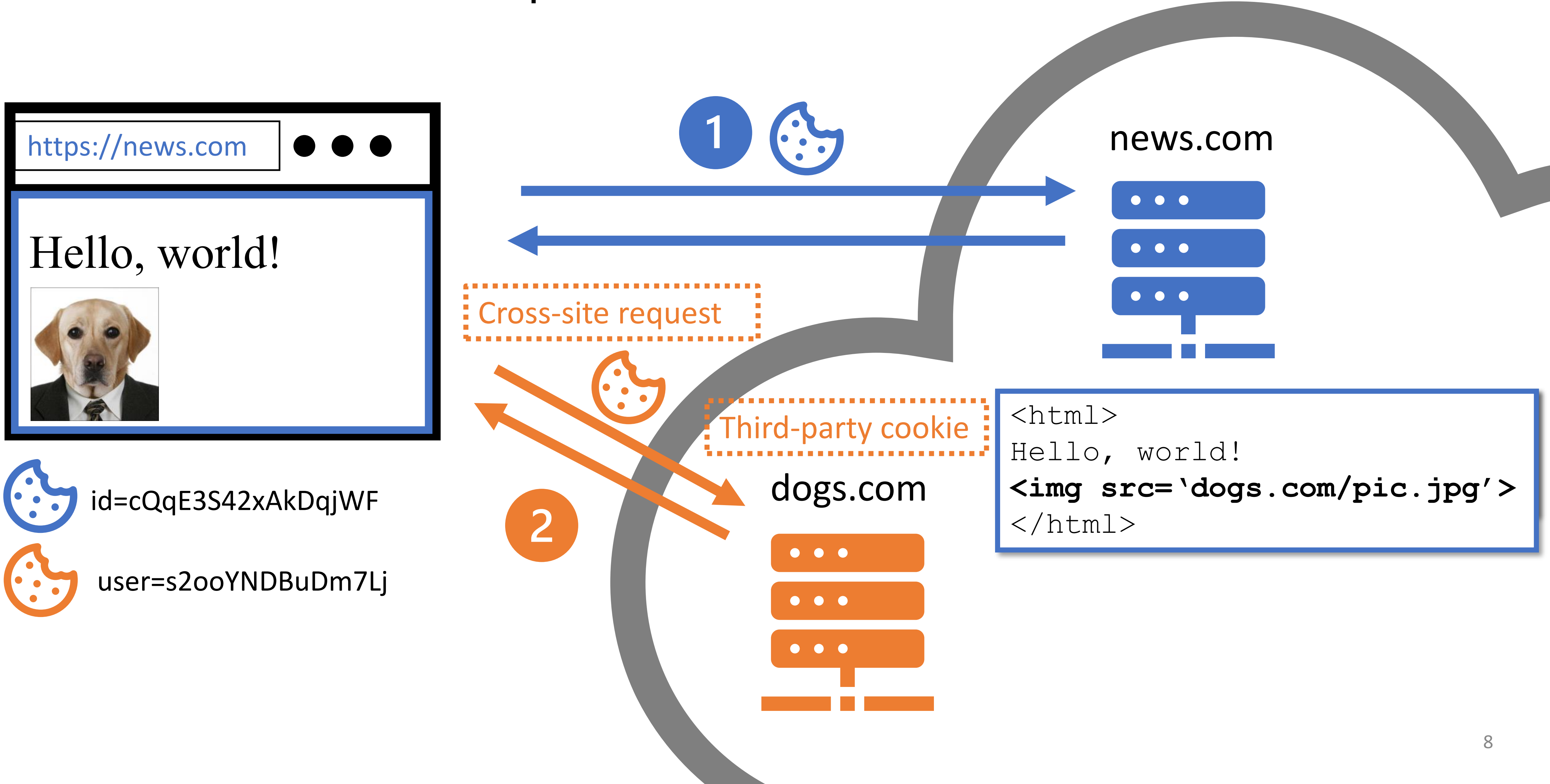


This Book Is Totally Safe 1st Edition

by Hacker (Author)

1.4 ★☆☆☆☆ 1,396 ratings See all formats and editions

Web 101: cookies and requests



https://news.com

Hello, world!

1

news.com

Cross-site request

Third-party cookie

2

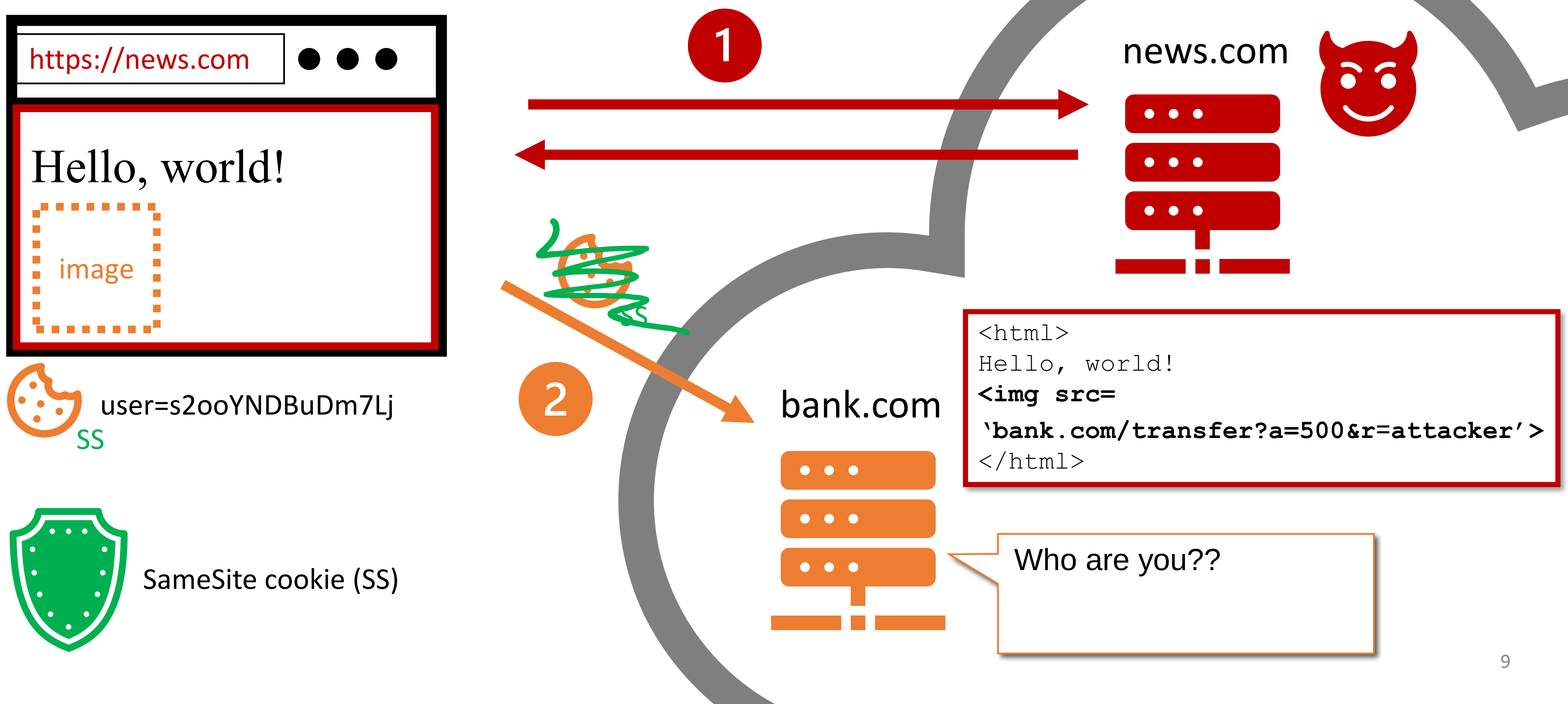
dogs.com

```
<html>
Hello, world!
<img src='dogs.com/pic.jpg'>
</html>
```

id=cQqE3S42xAkDqjWF

user=s2ooYNDBuDm7Lj

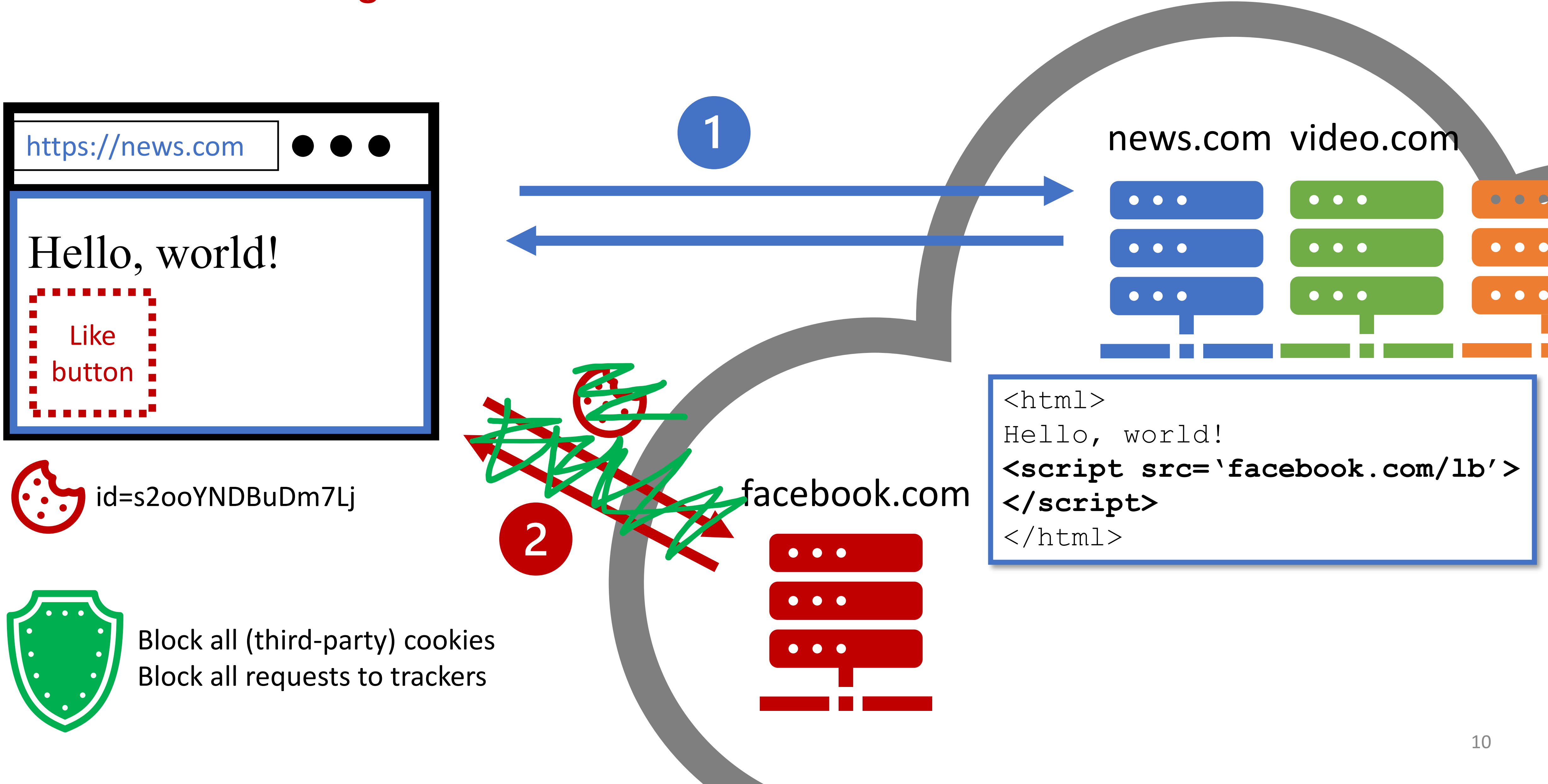
Cross-site Request Forgery (CSRF)



```
<html>
Hello, world!
<img src=
'bank.com/transfer?a=500&r=attacker'>
</html>
```

SameSite cookie (SS)

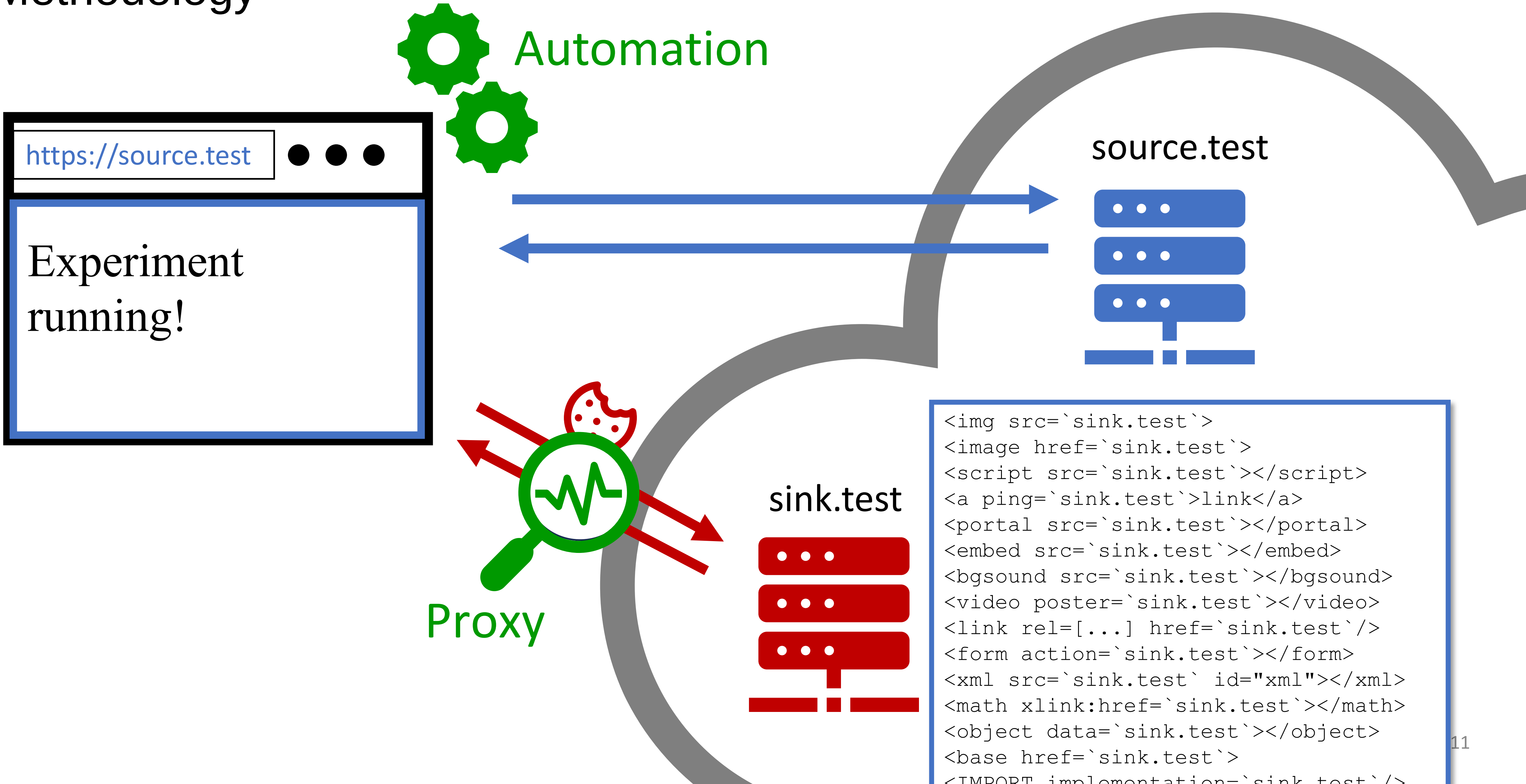
Cross-site tracking



```
<html>
Hello, world!
<script src='facebook.com/lb'>
</script>
</html>
```

Block all (third-party) cookies
Block all requests to trackers

Methodology



```
<img src=`sink.test`>
<image href=`sink.test`>
<script src=`sink.test`></script>
<a ping=`sink.test`>link</a>
<portal src=`sink.test`></portal>
<embed src=`sink.test`></embed>
<bgsound src=`sink.test`></bgsound>
<video poster=`sink.test`></video>
<link rel=[...] href=`sink.test`/>
<form action=`sink.test`></form>
<xml src=`sink.test` id="xml"></xml>
<math xlink:href=`sink.test`></math>
<object data=`sink.test`></object>
<base href=`sink.test`>
<IMPORT implementation=`sink.test`/>
```

46 **ad blocking** and **anti-tracking** extensions

Takeaways I

```
if cookies_disabled():
    request.exclude_cookie()
```

On / off switch – **conceptually simple**

Just oversights? 🤔

❌ 1 year later: only a few complete fixes

Many policies are **retroactively** added

New policy

↓

Scrutinize all supported features

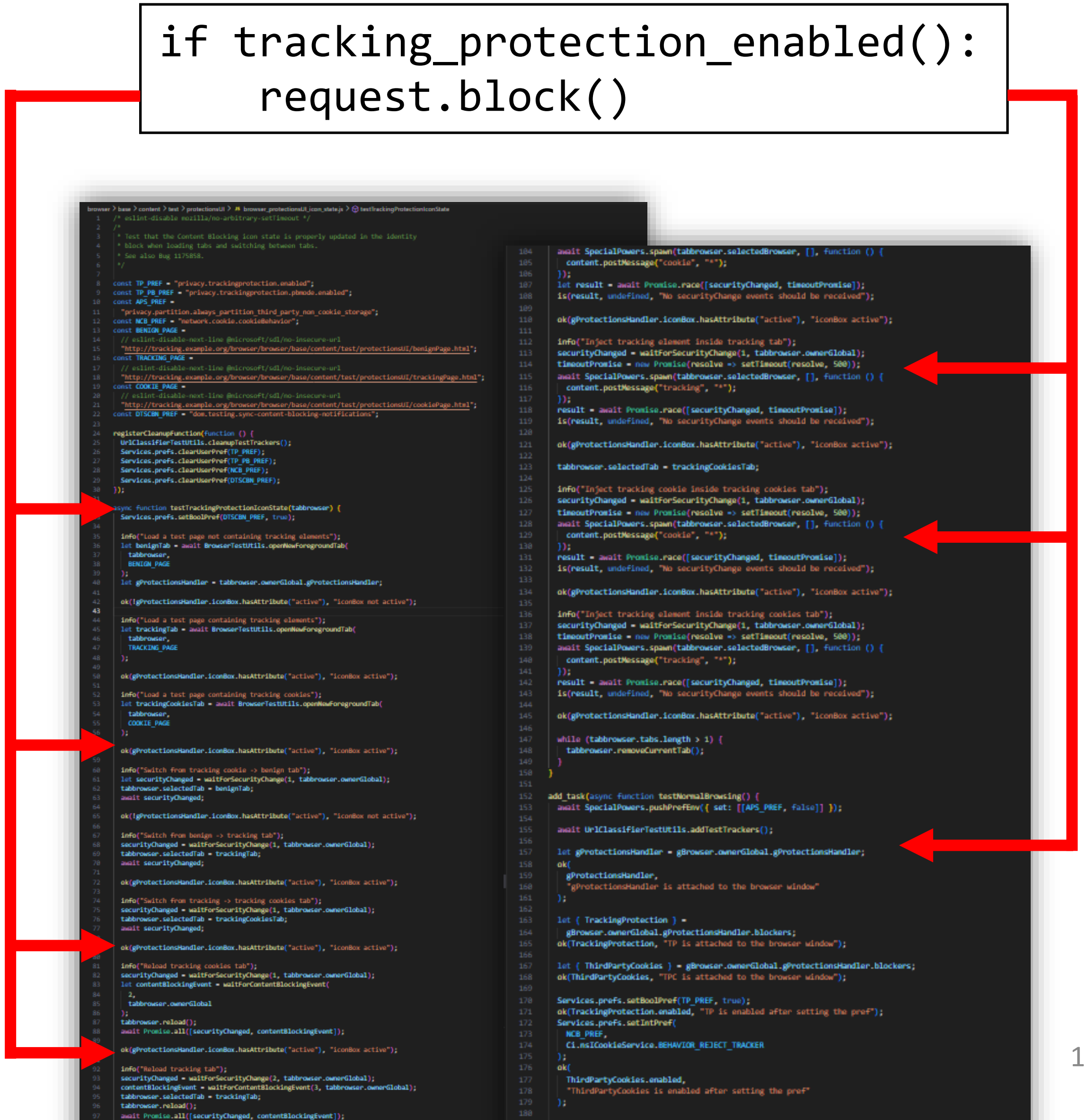
Update all relevant policies

↑

New feature

*

*10 – 100 features per release



```
if tracking_protection_enabled():
    request.block()
```

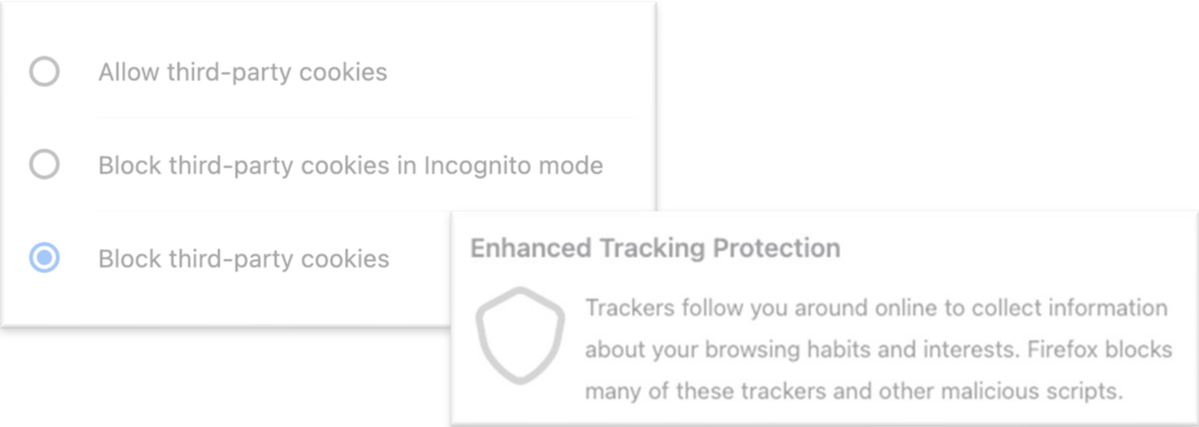
browser > base > content > test > protectionsUI > browser_protectionsUI_icon_state.js > testTrackingProtectionIconState

```
/* eslint-disable mozilla/no-arbitrary-setTimeout */
/*
 * Test that the Content Blocking icon state is properly updated in the identity
 * block when loading tabs and switching between tabs.
 * See also Bug 1175858.
 */

const TP_PREF = "privacy.trackingprotection.enabled";
const TP_PB_PREF = "privacy.trackingprotection.pbmode.enabled";
const APS_PREF =
  "privacy.partition.always_partition_third_party_non_cookie_storage";
const NCB_PREF = "network.cookie.cookieBehavior";
const BENIGN_PAGE =
  // eslint-disable-next-line @microsoft/sdl/no-insecure-url
  "http://tracking.example.org/browser/browser/base/content/test/protectionsUI/benignPage.html";
const TRACKING_PAGE =
  // eslint-disable-next-line @microsoft/sdl/no-insecure-url
  "http://tracking.example.org/browser/browser/base/content/test/protectionsUI/trackingPage.html";
const COOKIE_PAGE =
  // eslint-disable-next-line @microsoft/sdl/no-insecure-url
  "http://tracking.example.org/browser/browser/base/content/test/protectionsUI/cookiePage.html";
const DTSCBN_PREF = "dom.testing.sync-content-blocking-notifications";

registerCleanupFunction(function () {
  UrlClassifierTestUtils.cleanupTestTrackers();
  Services.prefs.clearUserPref(TP_PREF);
  Services.prefs.clearUserPref(TP_PB_PREF);
  Services.prefs.clearUserPref(NCB_PREF);
  Services.prefs.clearUserPref(DTSCBN_PREF);
});

async function testTrackingProtectionIconState(tabbrowser) {
  Services.prefs.setBoolPref(DTSCBN_PREF, true);

  info("Load a test page not containing tracking elements");
  let benignTab = await BrowserTestUtils.openNewForegroundTab(
    tabbrowser,
    BENIGN_PAGE
  );
  let gProtectionsHandler = tabbrowser.ownerGlobal.gProtectionsHandler;

  ok(!gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox not active");

  info("Load a test page containing tracking elements");
  let trackingTab = await BrowserTestUtils.openNewForegroundTab(
    tabbrowser,
    TRACKING_PAGE
  );

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Load a test page containing tracking cookies");
  let trackingCookiesTab = await BrowserTestUtils.openNewForegroundTab(
    tabbrowser,
    COOKIE_PAGE
  );

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Switch from tracking cookie -> benign tab");
  let securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  tabbrowser.selectedTab = benignTab;
  await securityChanged;

  ok(!gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox not active");

  info("Switch from benign -> tracking tab");
  securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  tabbrowser.selectedTab = trackingTab;
  await securityChanged;

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Switch from tracking -> tracking cookies tab");
  securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  tabbrowser.selectedTab = trackingCookiesTab;
  await securityChanged;

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Reload tracking cookies tab");
  securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  let contentBlockingEvent = waitForContentBlockingEvent(
    2,
    tabbrowser.ownerGlobal
  );
  tabbrowser.reload();
  await Promise.all([securityChanged, contentBlockingEvent]);

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Reload tracking tab");
  securityChanged = waitForSecurityChange(2, tabbrowser.ownerGlobal);
  contentBlockingEvent = waitForContentBlockingEvent(3, tabbrowser.ownerGlobal);
  tabbrowser.selectedTab = trackingTab;
  tabbrowser.reload();
  await Promise.all([securityChanged, contentBlockingEvent]);
```

```
  await SpecialPowers.spawn(tabbrowser.selectedBrowser, [], function () {
    content.postMessage("cookie", "*");
  });
  let result = await Promise.race([securityChanged, timeoutPromise]);
  is(result, undefined, "No securityChange events should be received");

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Inject tracking element inside tracking tab");
  securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  timeoutPromise = new Promise(resolve => setTimeout(resolve, 500));
  await SpecialPowers.spawn(tabbrowser.selectedBrowser, [], function () {
    content.postMessage("tracking", "*");
  });
  result = await Promise.race([securityChanged, timeoutPromise]);
  is(result, undefined, "No securityChange events should be received");

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  tabbrowser.selectedTab = trackingCookiesTab;

  info("Inject tracking cookie inside tracking cookies tab");
  securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  timeoutPromise = new Promise(resolve => setTimeout(resolve, 500));
  await SpecialPowers.spawn(tabbrowser.selectedBrowser, [], function () {
    content.postMessage("cookie", "*");
  });
  result = await Promise.race([securityChanged, timeoutPromise]);
  is(result, undefined, "No securityChange events should be received");

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  info("Inject tracking element inside tracking cookies tab");
  securityChanged = waitForSecurityChange(1, tabbrowser.ownerGlobal);
  timeoutPromise = new Promise(resolve => setTimeout(resolve, 500));
  await SpecialPowers.spawn(tabbrowser.selectedBrowser, [], function () {
    content.postMessage("tracking", "*");
  });
  result = await Promise.race([securityChanged, timeoutPromise]);
  is(result, undefined, "No securityChange events should be received");

  ok(gProtectionsHandler.iconBox.hasAttribute("active"), "iconBox active");

  while (tabbrowser.tabs.length > 1) {
    tabbrowser.removeCurrentTab();
  }
}

add_task(async function testNormalBrowsing() {
  await SpecialPowers.pushPrefEnv({ set: [[APS_PREF, false]] });

  await UrlClassifierTestUtils.addTestTrackers();

  let gProtectionsHandler = gBrowser.ownerGlobal.gProtectionsHandler;
  ok(
    gProtectionsHandler,
    "gProtectionsHandler is attached to the browser window"
  );

  let { TrackingProtection } =
    gBrowser.ownerGlobal.gProtectionsHandler.blockers;
  ok(TrackingProtection, "TP is attached to the browser window");

  let { ThirdPartyCookies } = gBrowser.ownerGlobal.gProtectionsHandler.blockers;
  ok(ThirdPartyCookies, "TPC is attached to the browser window");

  Services.prefs.setBoolPref(TP_PREF, true);
  ok(TrackingProtection.enabled, "TP is enabled after setting the pref");
  Services.prefs.setIntPref(
    NCB_PREF,
    Ci.nsICookieService.BEHAVIOR_REJECT_TRACKER
  );
  ok(
    ThirdPartyCookies.enabled,
    "ThirdPartyCookies is enabled after setting the pref"
  );
```

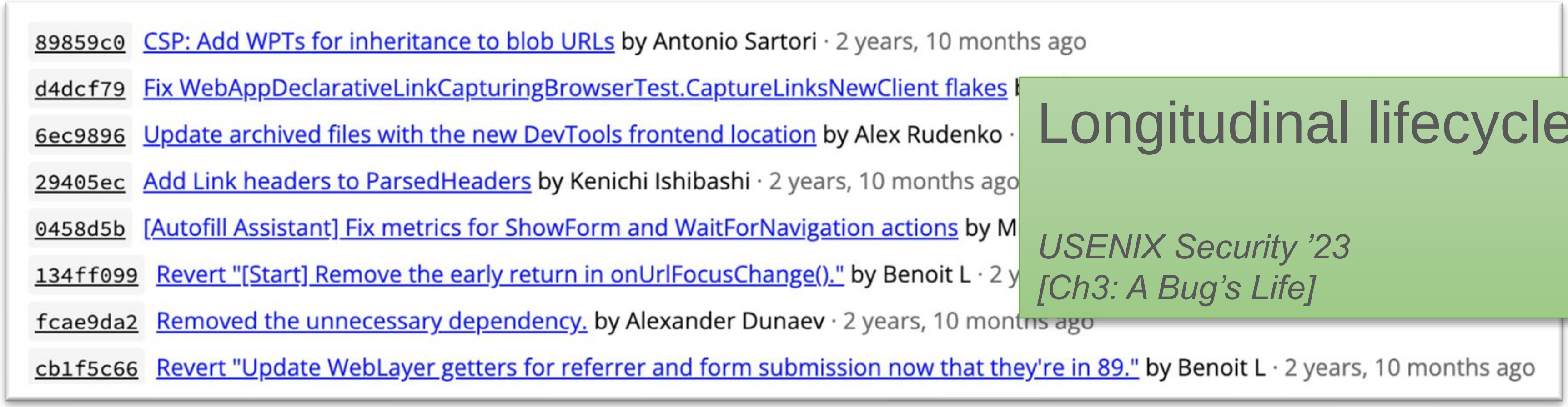
Evaluation of cookie and request policies

USENIX Security ‘18
[Ch2: Who Left Open the Cookie Jar?]


○ Allow third-party cookies
○ Block third-party cookies in Incognito mode
● Block third-party cookies

Enhanced Tracking Protection

Trackers follow you around online to collect information about your browsing habits and interests. Firefox blocks many of these trackers and other malicious scripts.

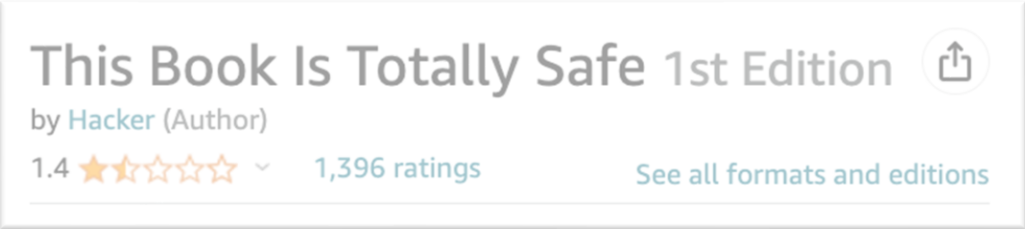

89859c0 CSP: Add WPTs for inheritance to blob URLs by Antonio Sartori · 2 years, 10 months ago
d4dcf79 Fix WebAppDeclarativeLinkCapturingBrowserTest.CaptureLinksNewClient flakes b
6ec9896 Update archived files with the new DevTools frontend location by Alex Rudenko ·
29405ec Add Link headers to ParsedHeaders by Kenichi Ishibashi · 2 years, 10 months ago
0458d5b [Autofill Assistant] Fix metrics for ShowForm and WaitForNavigation actions by M
134ff099 Revert "[Start] Remove the early return in onUrlFocusChange()." by Benoit L · 2 y
fcae9da2 Removed the unnecessary dependency. by Alexander Dunaev · 2 years, 10 months ago
cb1f5c66 Revert "Update WebLayer getters for referrer and form submission now that they're in 89." by Benoit L · 2 years, 10 months ago

Longitudinal lifecycle analysis of CSP bugs

USENIX Security ’23
[Ch3: A Bug’s Life]

Security and privacy in EPUB reading systems

IEEE S&P ’21
[Ch4: Reading Between the Lines]


This Book Is Totally Safe 1st Edition

by Hacker (Author)

1.4 ★☆☆☆☆ 1,396 ratings

See all formats and editions

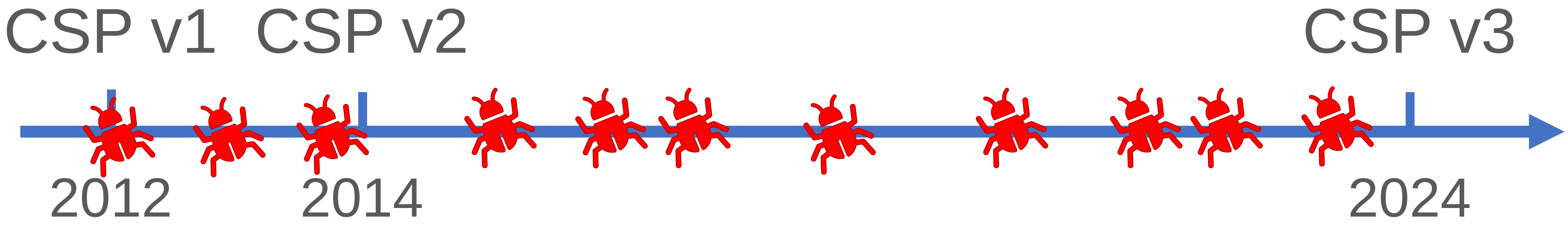
Content Security Policy (CSP)

- **Defense in-depth** against **content injection attacks** (e.g., XSS) and **clickjacking**
  - Defined by website
  - Enforced by web browser

- “**Living** standard”

Content Security Policy Level 3

W3C Working Draft, 23 January 2024

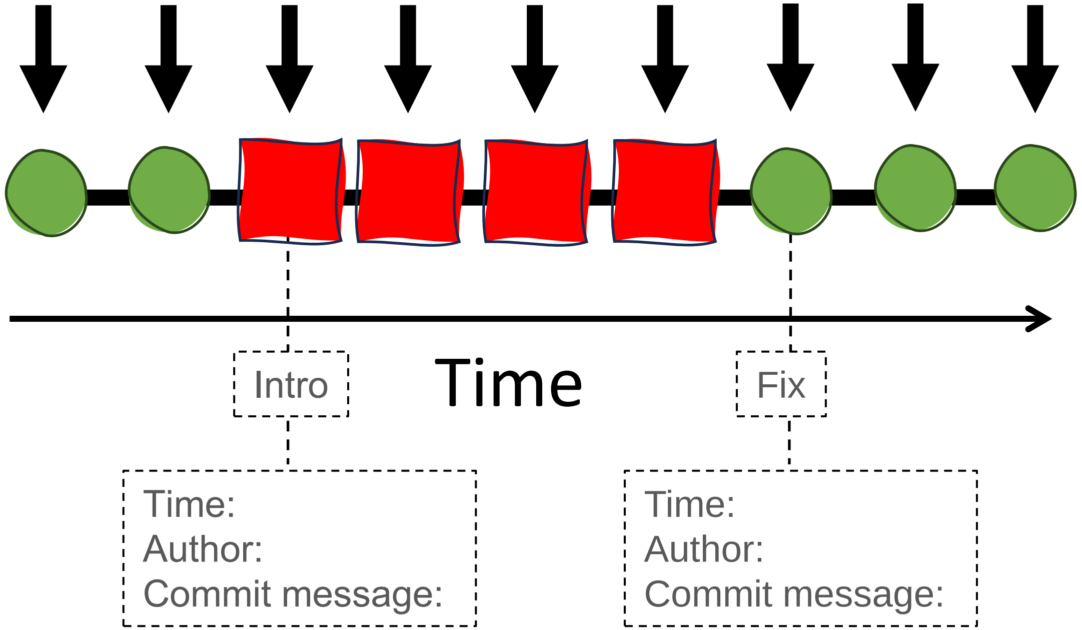


CSP v1 CSP v2 CSP v3

2012 2014 2024

Code revisions

Version Control System (e.g., Git)



Time:
Author:
Commit message:

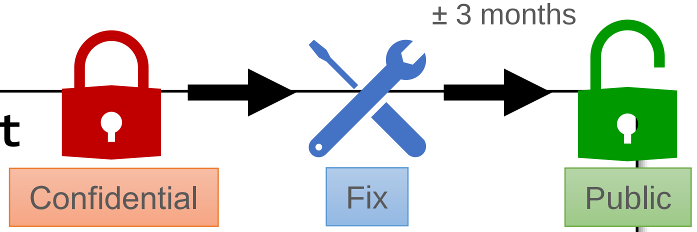
Time:
Author:
Commit message:

⚠ **> 1.000.000 revisions**

⚠ **> 100 revisions / day**

```
say_no(a: str) -> str:
    return `no`

if __name__ == `__main__`:
    while True:
        arg = input()
        if arg == `exit`:
            exit(0)
        r = say_no(arg)
        print(f`computer says {r}`)
        exit(0)
```

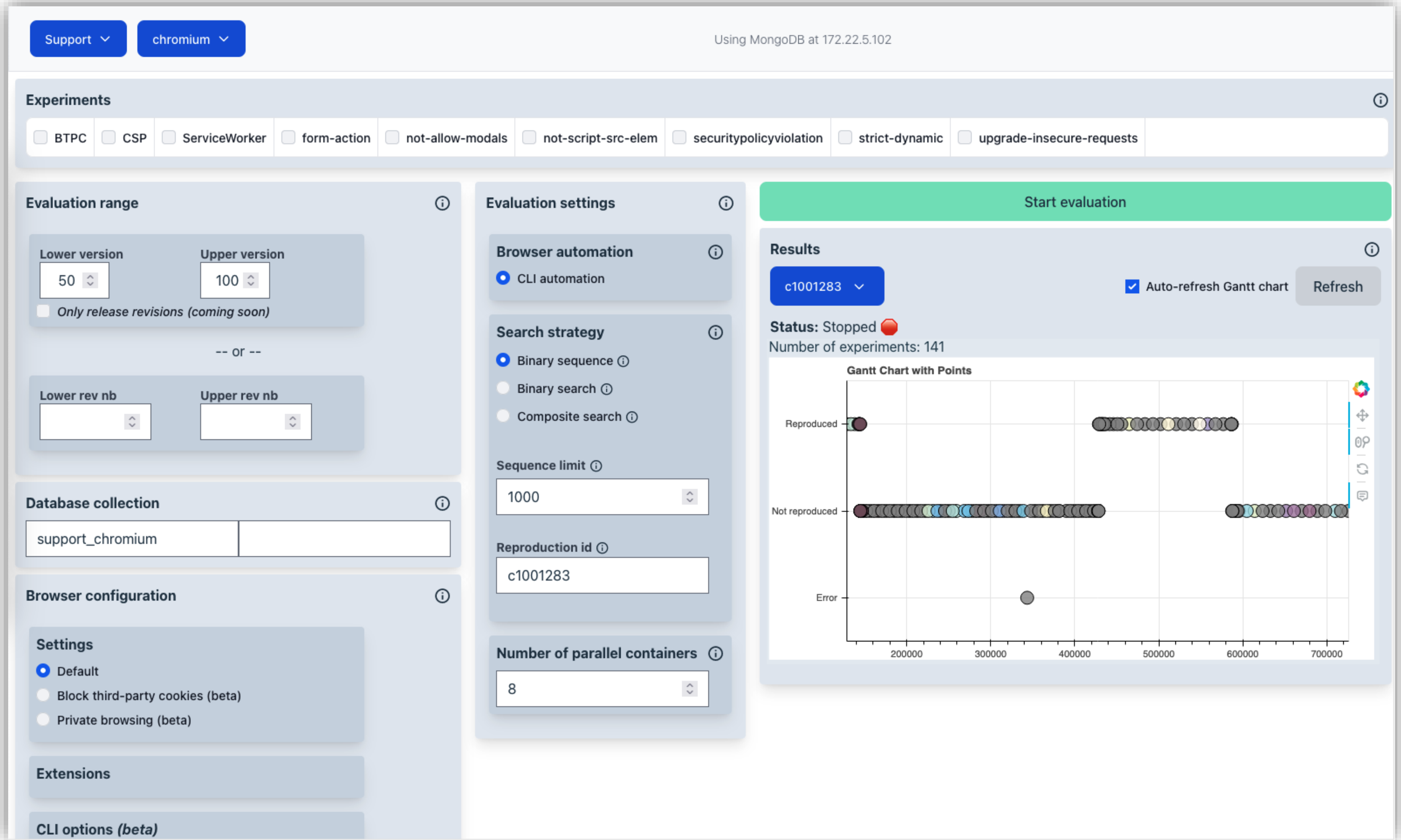
Proof of Concept (PoC)



```
Reproduce bug:
1. Open browser
2. Visit index.html
3. ...
```

BugHog

Automated lifecycle pinpointing



Experiments

□ BTPC □ CSP □ ServiceWorker □ form-action □ not-allow-modals □ not-script-src-elem □ securitypolicyviolation □ strict-dynamic □ upgrade-insecure-requests

Evaluation range

Lower version: 50 | Upper version: 100

□ *Only release revisions (coming soon)*

-- or --

Lower rev nb: | Upper rev nb:

Database collection

support_chromium

Browser configuration

Settings

● Default
○ Block third-party cookies (beta)
○ Private browsing (beta)

Extensions

CLI options *(beta)*

Evaluation settings

Browser automation

● CLI automation

Search strategy

● Binary sequence
○ Binary search
○ Composite search

Sequence limit: 1000

Reproduction id: c1001283

Number of parallel containers: 8

Results

c1001283

☑ Auto-refresh Gantt chart

Status: Stopped 🔴

Number of experiments: 141

✓ PoCs of CSP bugs

→ 75 unique bugs

✓ Revision binaries

Delta with state of practice

+ Fully containerized
+ Dependencies managed

Chromium v25 – latest

Firefox v23 – latest

+ Concurrency

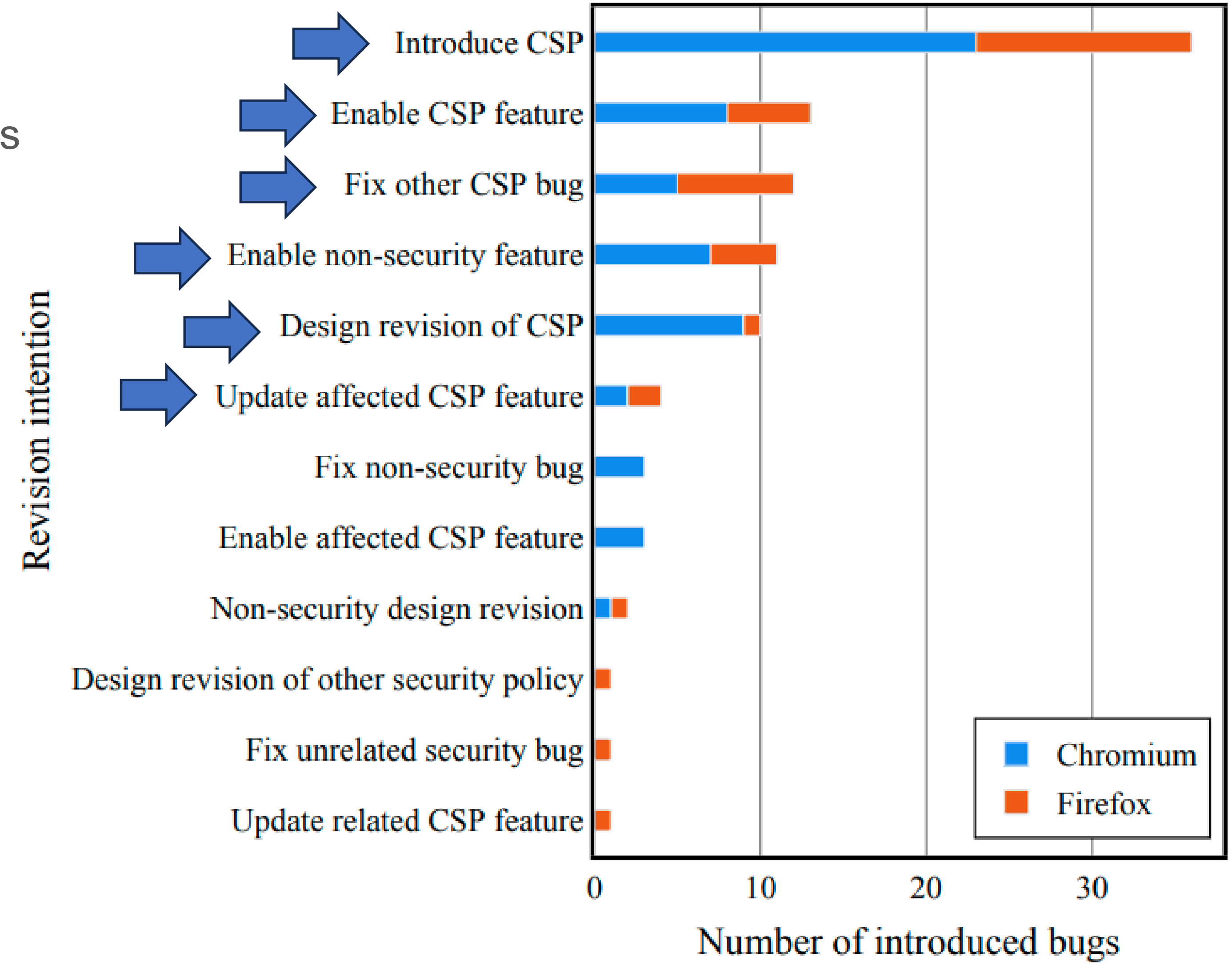
1. Bug introducing revisions

- Half of all bugs are **foundational**
    - $5000 bug lived under the radar for 8 years

- Modifications to **CSP logic** are likely to cause new bugs

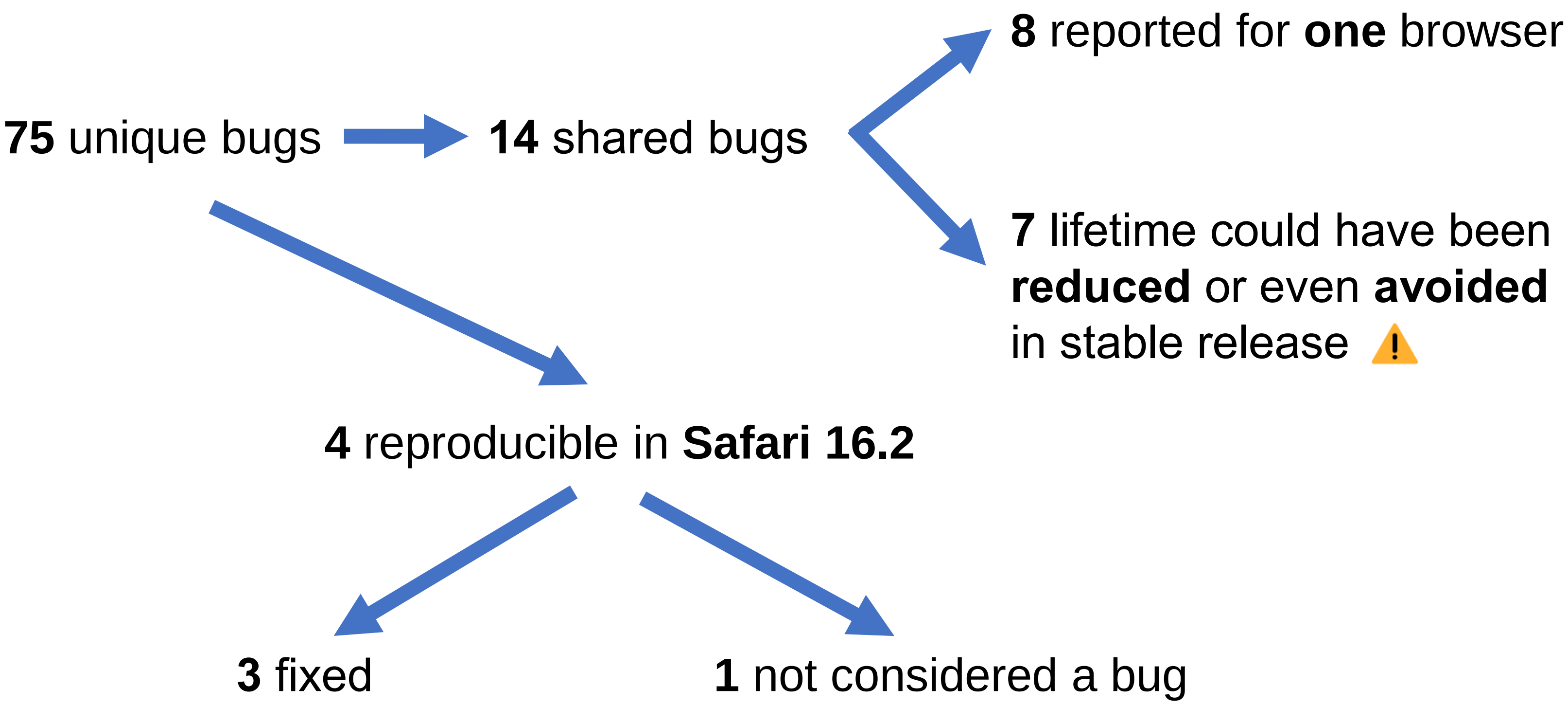
- **Non-security feature introductions** can act as bypass
    - **Fragmented enforcement** logic may lead to oversights

Intentions of bug introducing revisions



2. Room for improvement for cross-browser bug sharing

- Current practice: ***Web Platform Tests (WPT)***
  - Vendors push and pull regression tests to and from shared repo
- **Cross-browser** evaluation



75 unique bugs

14 shared bugs

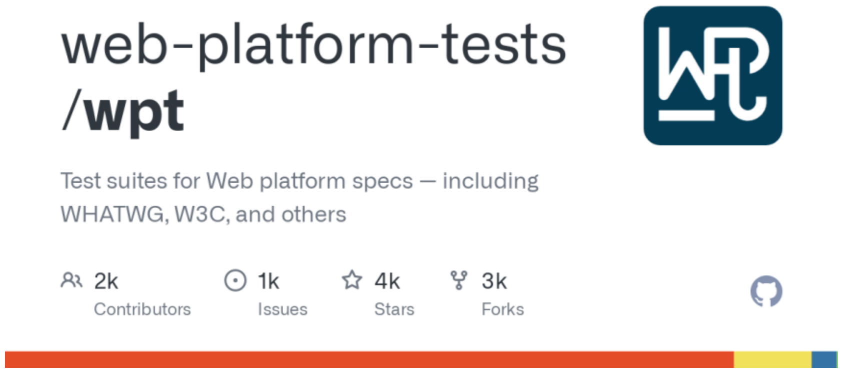
8 reported for **one** browser

7 lifetime could have been **reduced** or even **avoided** in stable release ⚠

4 reproducible in **Safari 16.2**

3 fixed

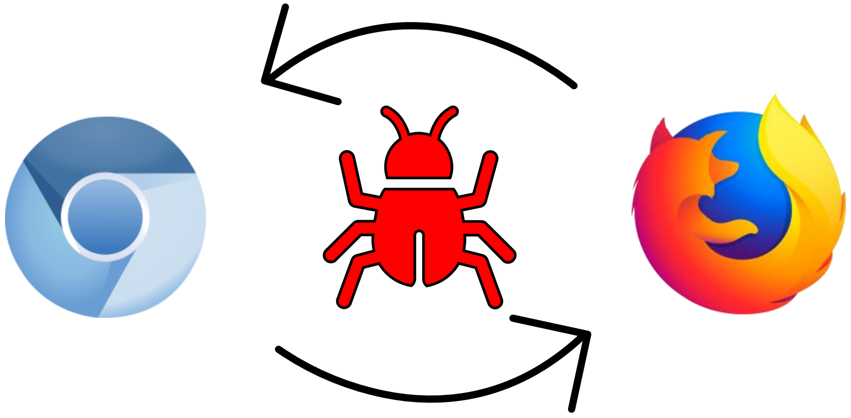
1 not considered a bug



web-platform-tests
/**wpt**

Test suites for Web platform specs — including WHATWG, W3C, and others

2k Contributors · 1k Issues · 4k Stars · 3k Forks



⚠ Safari was exposed for **> 1 year** for each of these bugs ⚠

3. Inconsistent bug handling can lead to premature disclosure

2 Chromium bugs

1 Firefox bug

> 1 year avoidable exposure

↓

Still present in the **latest release** at the time of the evaluation

✓ Reported and fixed

Takeaways II

❶ First longitudinal bug **lifecycle analysis**

- Based on **empirical evidence**
- **Independent** of developer labels

Bug **handling** and **sharing**

- Current practice leaves **room for improvement**
- **Avoidable** exposure

BugHog: **open-source**

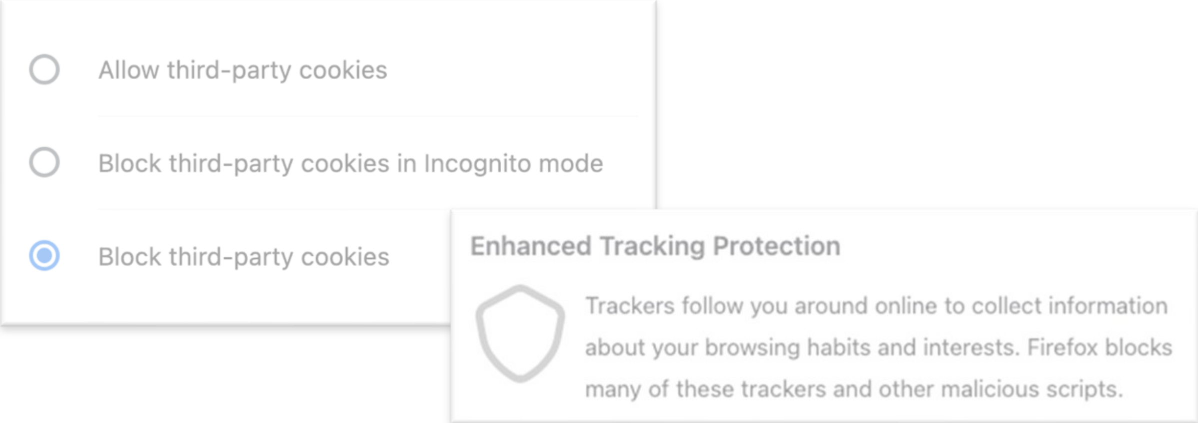
- Researchers and browser vendors
- From **policies** to **multi-stage attacks**

☠ **Premature disclosure** of bugs

You get an attack vector!
You get an attack vector!
Everybody gets a free attack vector!

Evaluation of cookie and request policies

USENIX Security ‘18
[Ch2: Who Left Open the Cookie Jar?]



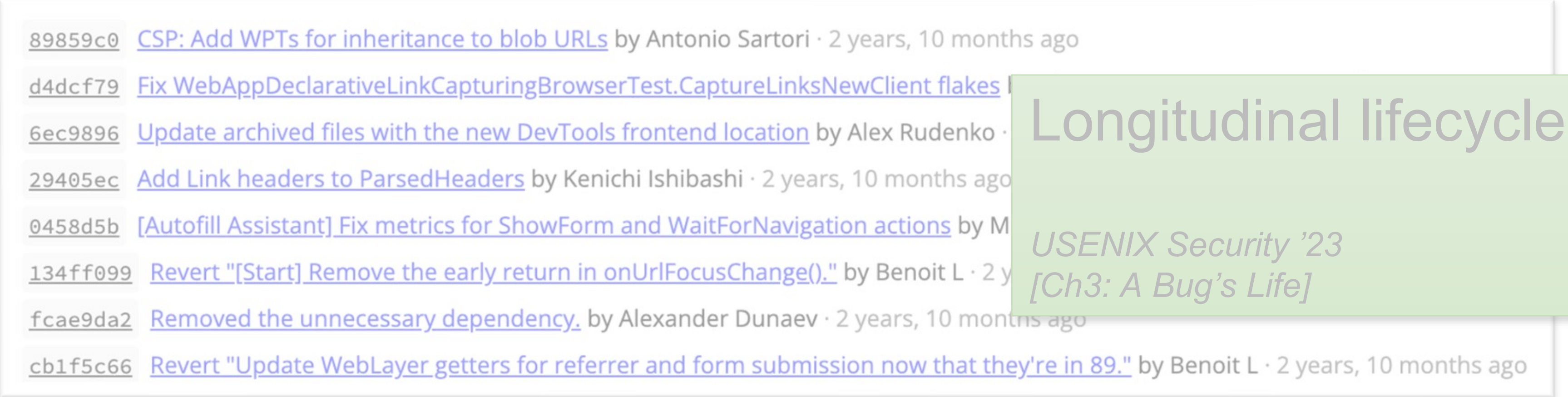
○ Allow third-party cookies

○ Block third-party cookies in Incognito mode

● Block third-party cookies

Enhanced Tracking Protection

Trackers follow you around online to collect information about your browsing habits and interests. Firefox blocks many of these trackers and other malicious scripts.



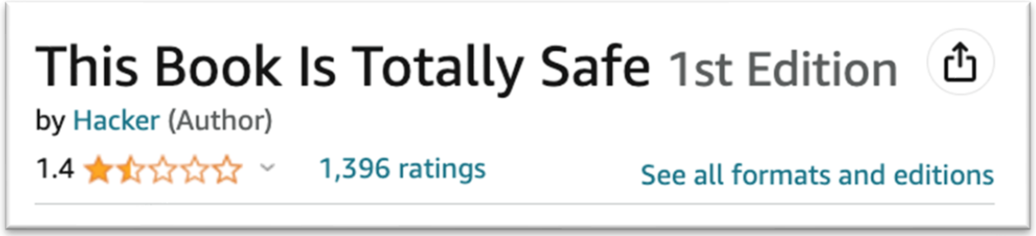
```
89859c0  CSP: Add WPTs for inheritance to blob URLs by Antonio Sartori · 2 years, 10 months ago
d4dcf79  Fix WebAppDeclarativeLinkCapturingBrowserTest.CaptureLinksNewClient flakes b
6ec9896  Update archived files with the new DevTools frontend location by Alex Rudenko ·
29405ec  Add Link headers to ParsedHeaders by Kenichi Ishibashi · 2 years, 10 months ago
0458d5b  [Autofill Assistant] Fix metrics for ShowForm and WaitForNavigation actions by M
134ff099 Revert "[Start] Remove the early return in onUrlFocusChange()." by Benoit L · 2 y
fcae9da2 Removed the unnecessary dependency. by Alexander Dunaev · 2 years, 10 months ago
cb1f5c66 Revert "Update WebLayer getters for referrer and form submission now that they're in 89." by Benoit L · 2 years, 10 months ago
```

Longitudinal lifecycle analysis of CSP bugs

USENIX Security ’23
[Ch3: A Bug’s Life]

Security and privacy in EPUB reading systems

IEEE S&P ’21
[Ch4: Reading Between the Lines]



This Book Is Totally Safe 1st Edition

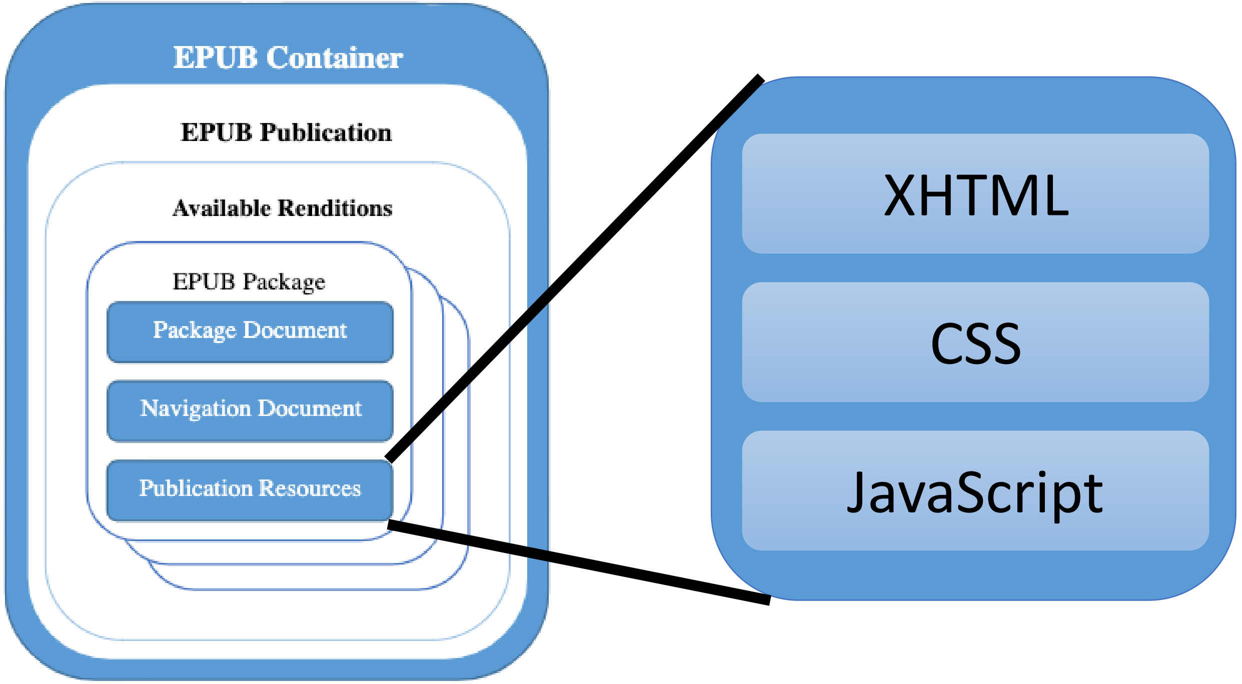
by Hacker (Author)

1.4 ★☆☆☆☆ 1,396 ratings See all formats and editions ePUB 3.2

.EPUB (.ZIP)

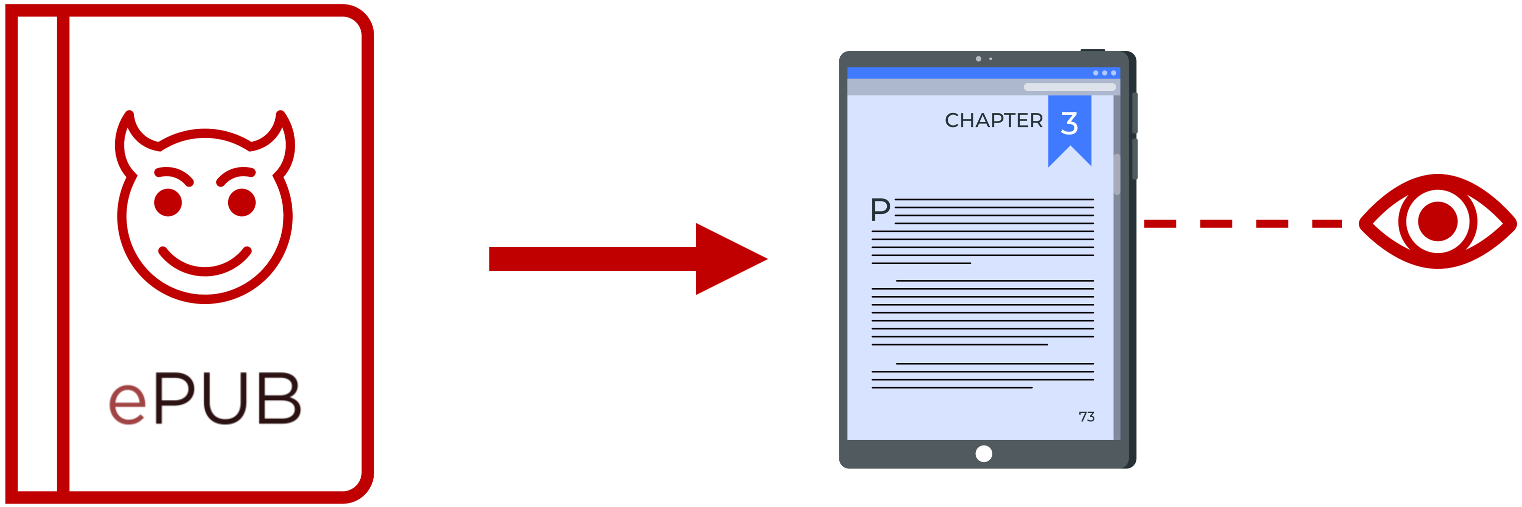
EPUB file

EPUB reading system



Lorem ipsum dolor sit amet, te quo modo petentium. Putent atomorum electram no usu. His iriure volumus hendrerit id, ea novum deserunt vel. Et sint graeco mel. Id ius detraxit incorrupte temporibus, sint inani vitae eos et, quod posidonium

Methodology



92 reading applications

iOS
android
macOS
Ubuntu

5 reading devices

ONYX
tolino
PocketBook
amazon kindle
kobo

Capabilities of malicious e-book

ePUB

Feature API access
5%

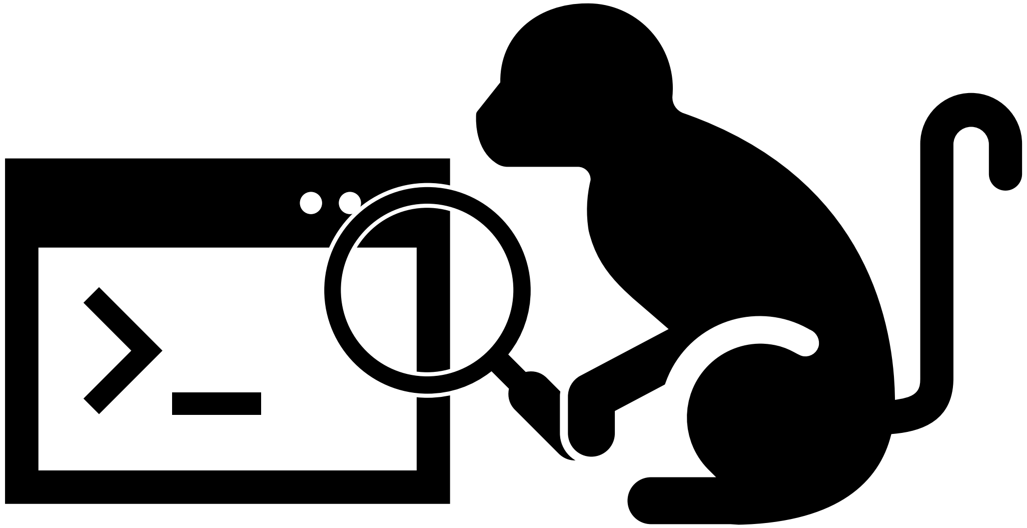
± 50% **untrusted JS execution** and **remote communication**

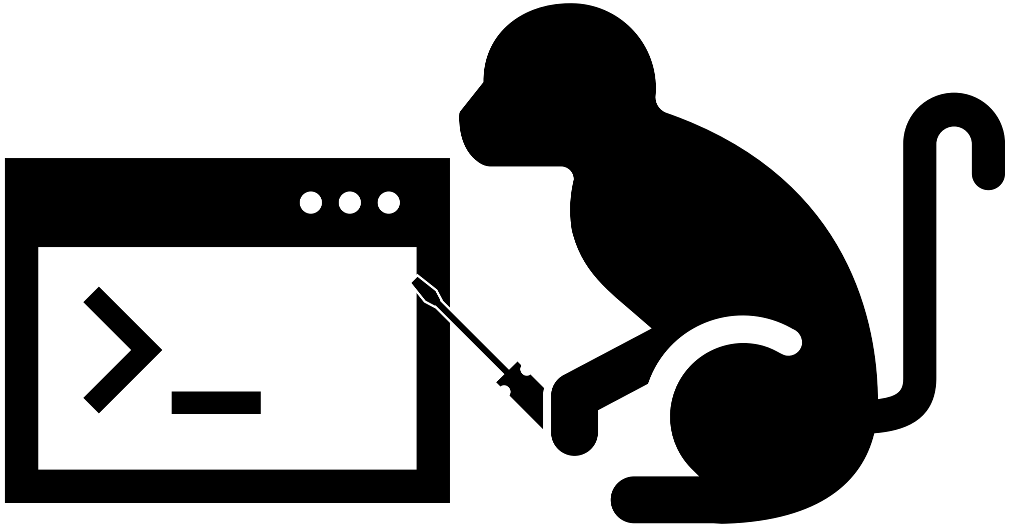
Opening apps through **URI handles**
25%

File system access
Existence: 16%
Steal contents: 8%

Insecure engines
2 / 5 e-readers

Amazon Kindle embedded a **10 year old WebKit** version





Case studies

Apple Books

→ persistent DOS
→ user information disclosure

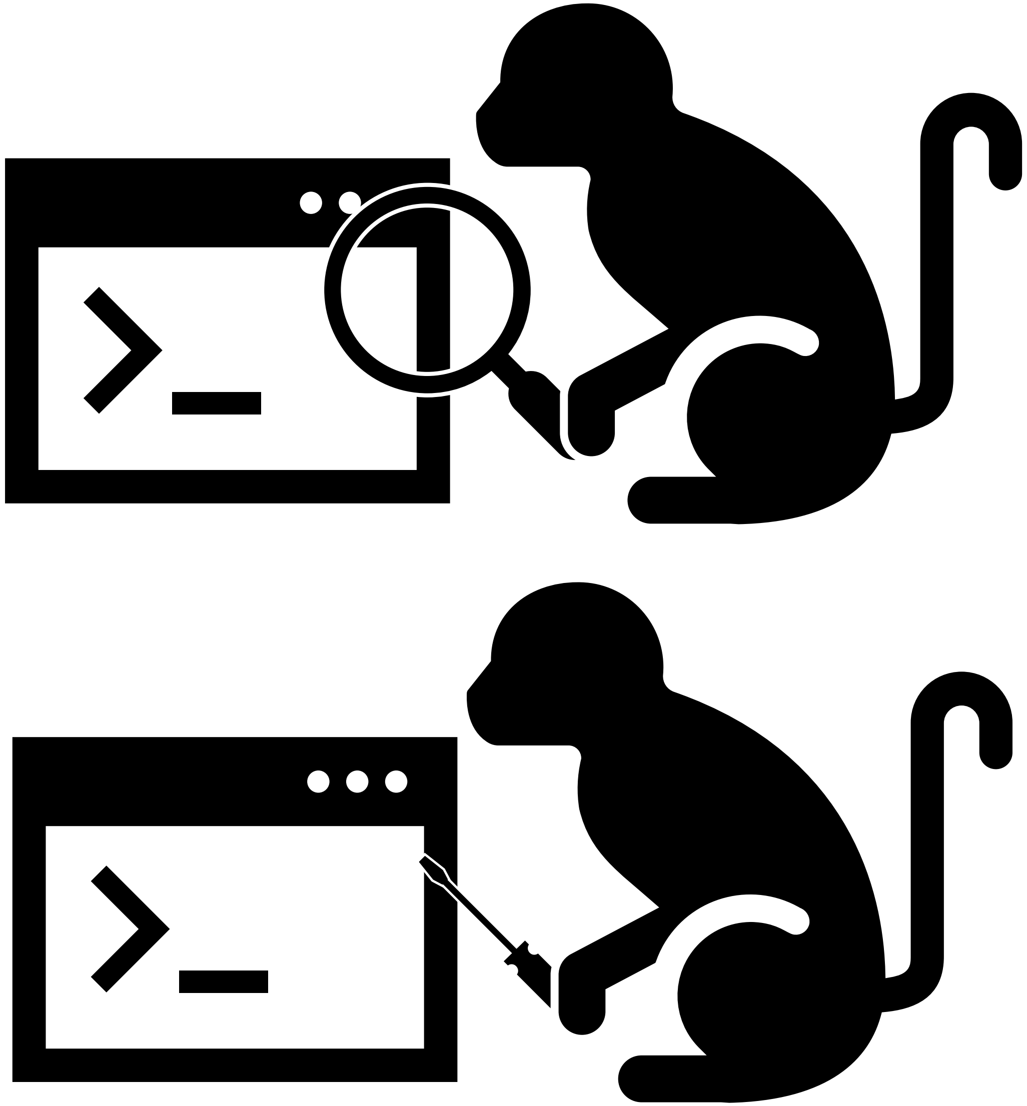
EPUBReader

→ universal XSS

Amazon Kindle

amazon kindle

→ information leaking



Capability (ab)use in the wild

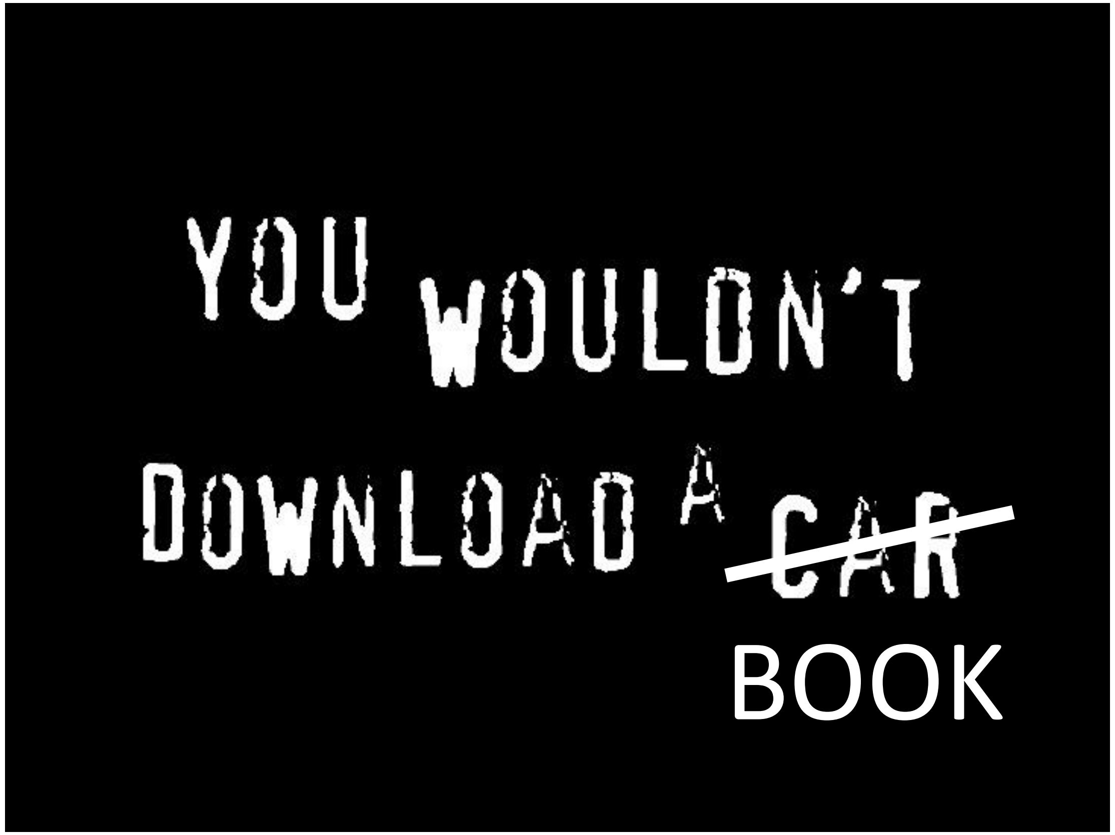
amazon books

Google Play Books

kobo

Project Gutenberg

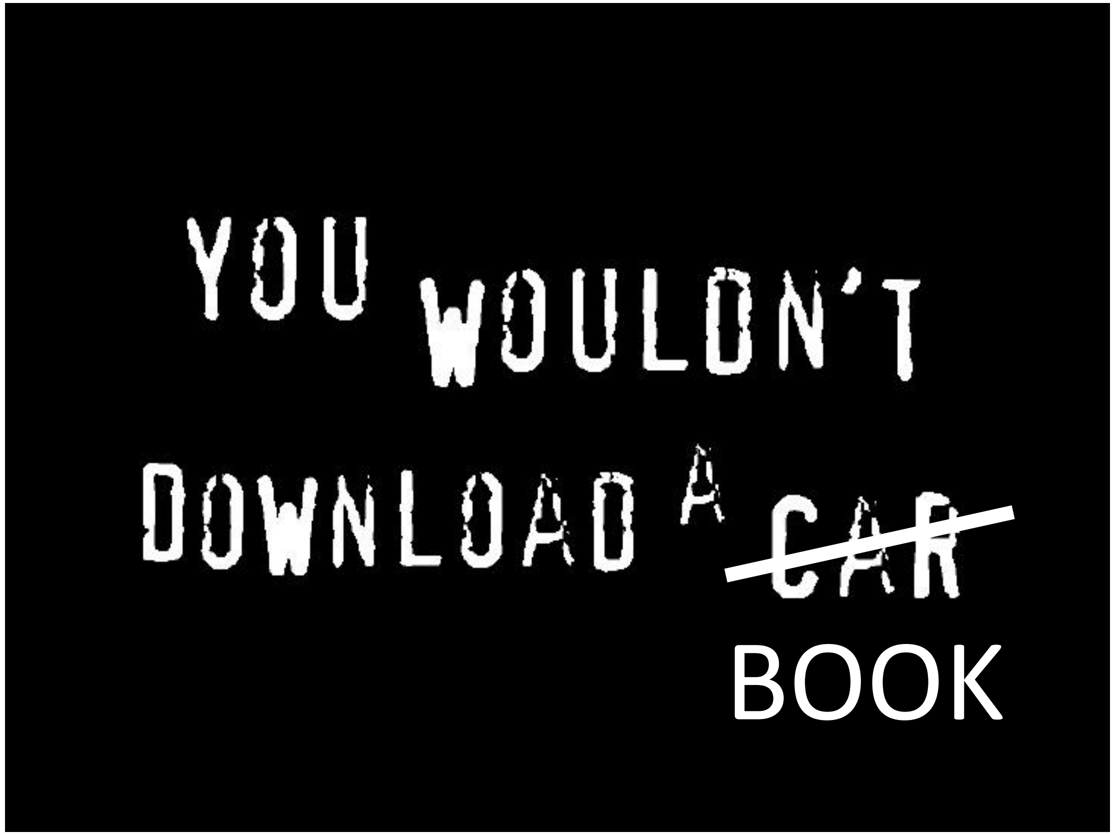
eBooks.com



Capability (ab)use in the wild

- Malicious EPUBs distributed through illegal channels
  - The Pirate Bay, 4shared
  - +/- 9,000 EPUBs

< 1% contained JavaScript (all benign)



Capability (ab)use in the wild

amazon books
Google Play Books
kobo
Project Gutenberg
eBooks.com

- Tracking EPUBs distributed through legal channels
  - Free e-books from the most popular EPUB vendors

↓

No indications of tracking

Are self-published EPUBs sufficiently sanitized?

*"This is not for everybody.
Like, really, actually for nobody."*

→

→

Publication

amazon books

Smashwords™
your ebook. your way.

Get it on
Apple Books

Rakuten kobo

94%

Google Play
Books

BARNES&NOBLE

Takeaways III

Embedding browser engine in **native applications**

- **Insecure** configurations
- Attack surface is **needlessly large**
- **Responsible disclosure** for 37 reading systems

Inadequate sanitization of self-published e-books

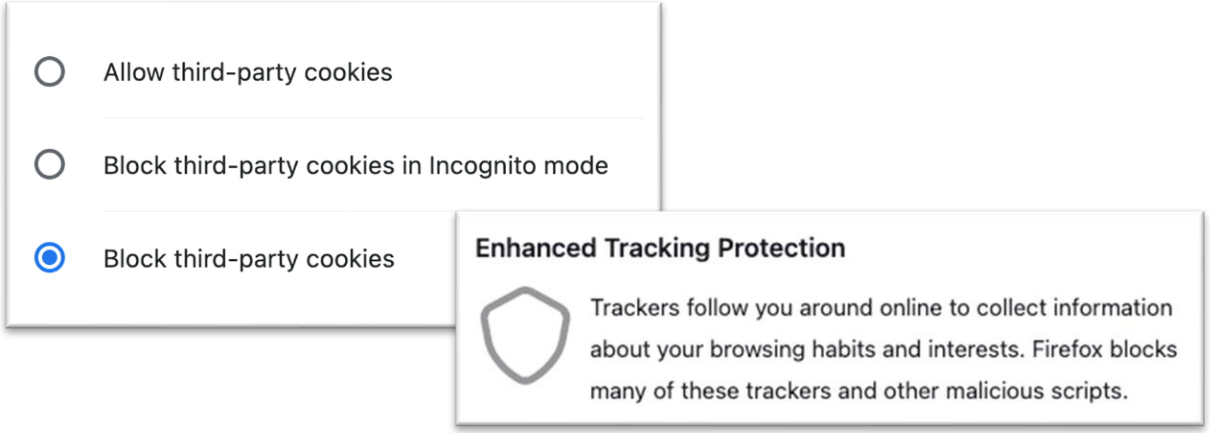
Collaboration with **W3C**

- Integration into the **official testbed**
- Improvements to the **standard**

Other notorious browser engine embedders

Evaluation of cookie and request policies

USENIX Security ‘18
[Ch2: Who Left Open the Cookie Jar?]



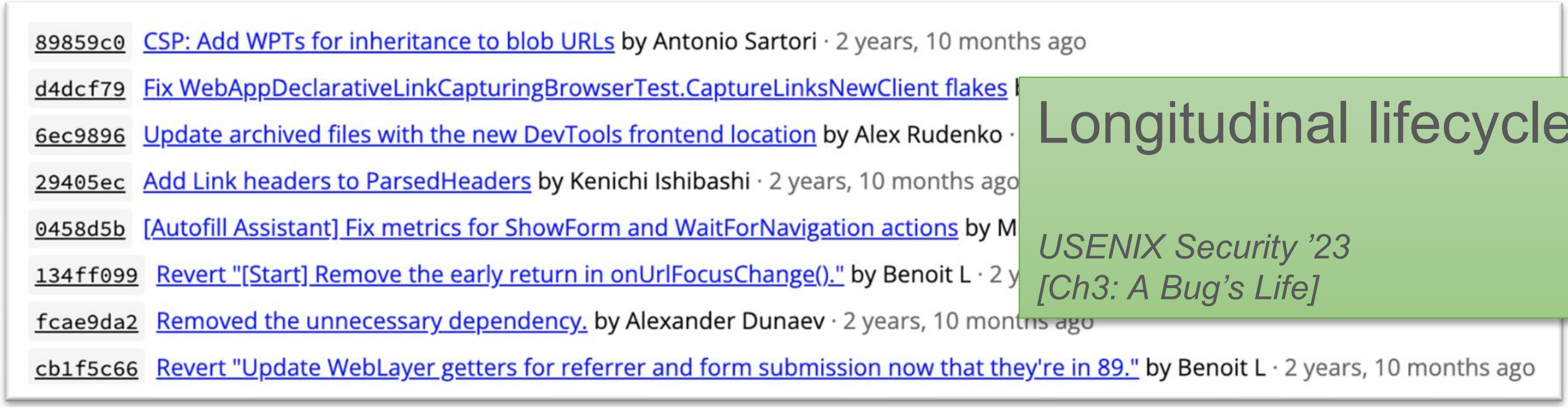
○ Allow third-party cookies

○ Block third-party cookies in Incognito mode

● Block third-party cookies

Enhanced Tracking Protection

Trackers follow you around online to collect information about your browsing habits and interests. Firefox blocks many of these trackers and other malicious scripts.



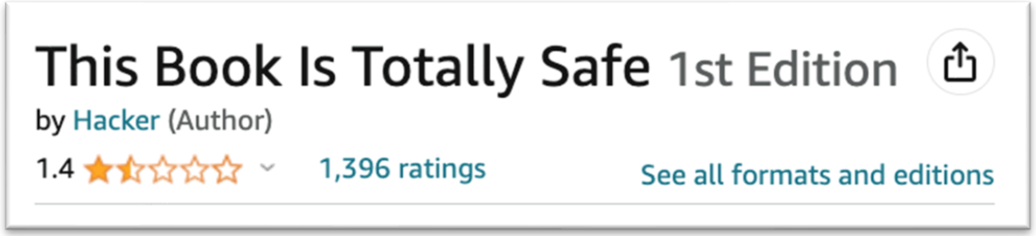
```
89859c0 CSP: Add WPTs for inheritance to blob URLs by Antonio Sartori · 2 years, 10 months ago
d4dcf79 Fix WebAppDeclarativeLinkCapturingBrowserTest.CaptureLinksNewClient flakes
6ec9896 Update archived files with the new DevTools frontend location by Alex Rudenko ·
29405ec Add Link headers to ParsedHeaders by Kenichi Ishibashi · 2 years, 10 months ago
0458d5b [Autofill Assistant] Fix metrics for ShowForm and WaitForNavigation actions by M
134ff099 Revert "[Start] Remove the early return in onUrlFocusChange()." by Benoit L · 2 y
fcae9da2 Removed the unnecessary dependency. by Alexander Dunaev · 2 years, 10 months ago
cb1f5c66 Revert "Update WebLayer getters for referrer and form submission now that they're in 89." by Benoit L · 2 years, 10 months ago
```

Longitudinal lifecycle analysis of CSP bugs

USENIX Security ’23
[Ch3: A Bug’s Life]

Security and privacy in EPUB reading systems

IEEE S&P ’21
[Ch4: Reading Between the Lines]



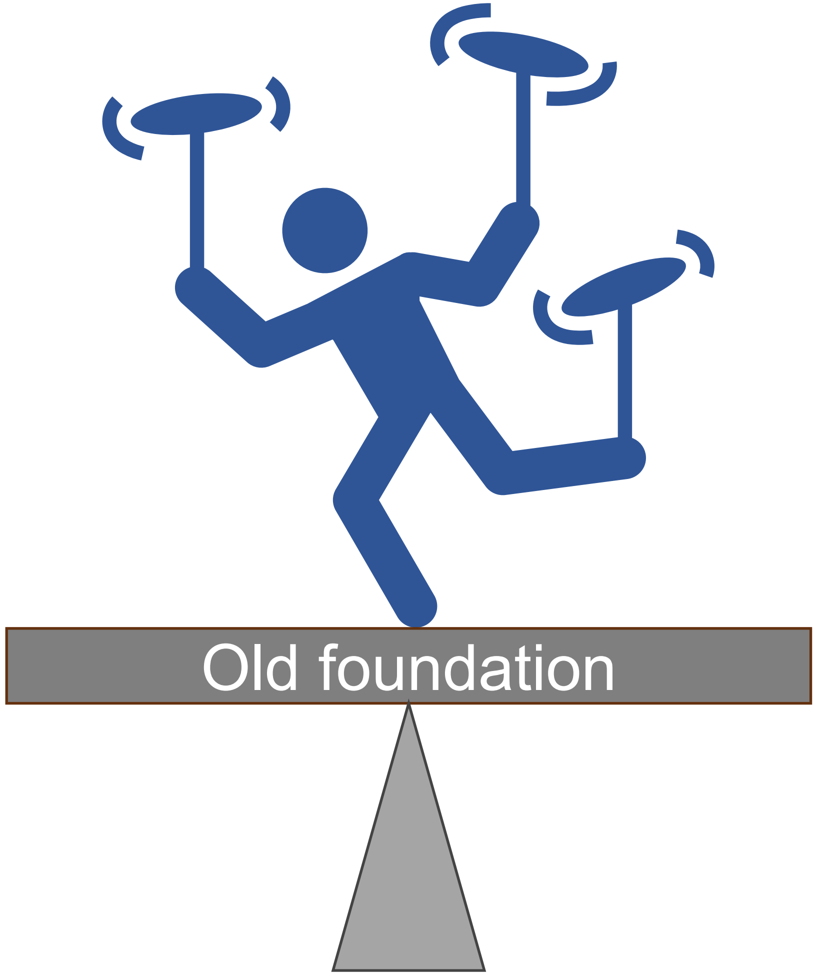
This Book Is Totally Safe 1st Edition

by Hacker (Author)

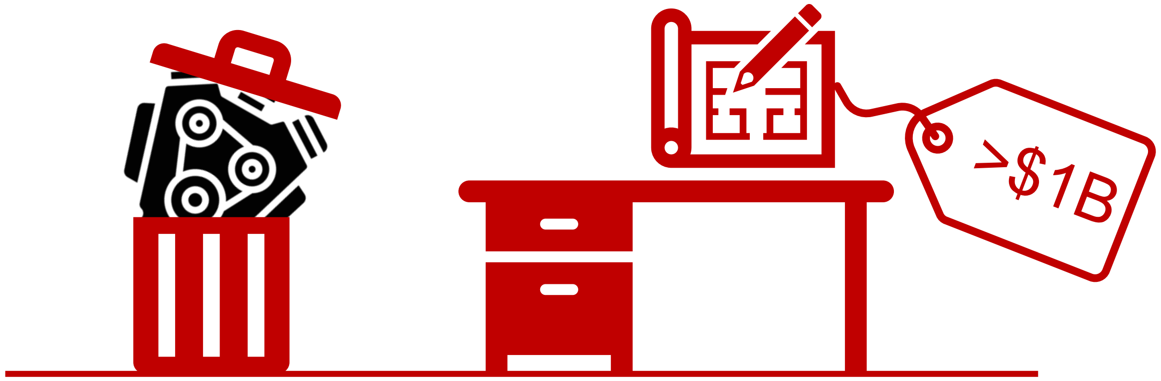
1.4 ★★☆☆☆ 1,396 ratings

See all formats and editions

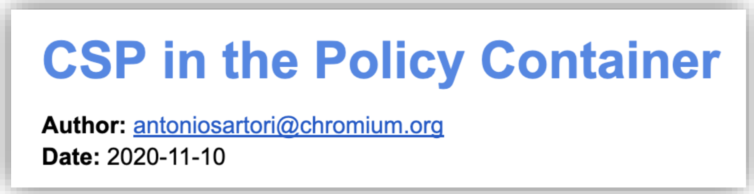
More features, more problems



Idealistic approach

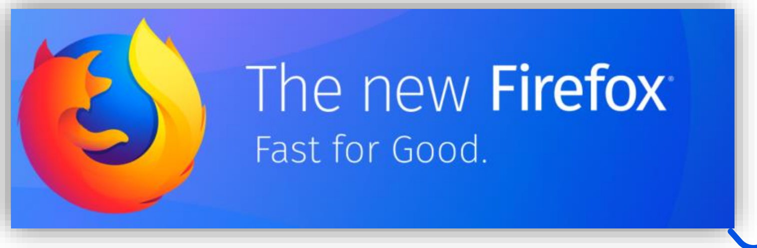


Pragmatic approach



CSP in the Policy Container

Author: antoniosartori@chromium.org
Date: 2020-11-10



+6%

Avoid exposure due to **bug handling** and **sharing issues**

- **Standardized** bug reporting language
- Shared bug **reporting platform**

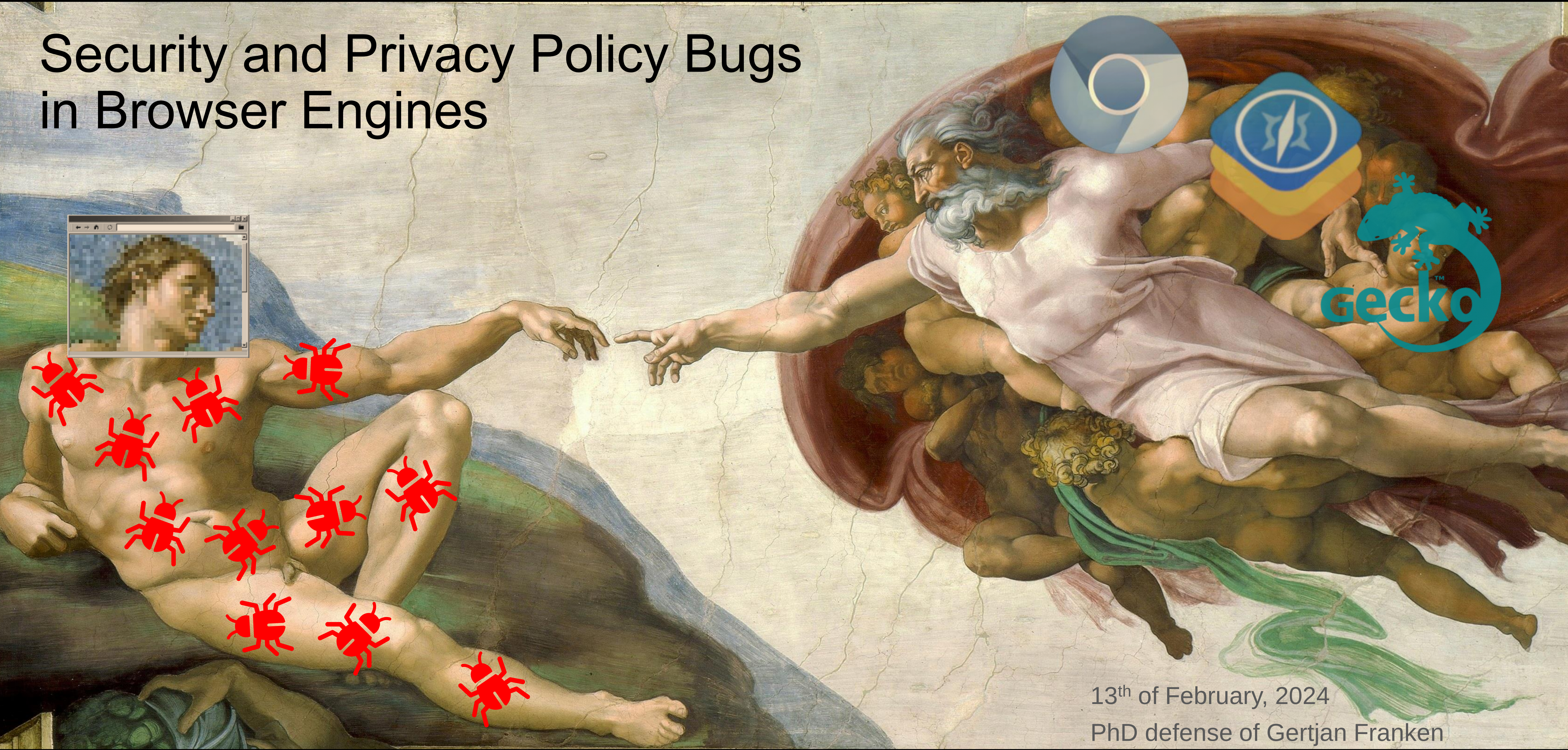
In depth bug analysis: **BugHog**!

Avoid **needlessly large attack surfaces** of embedded browser engines

Modular browser engines

Transparency towards developers and users

Security and Privacy Policy Bugs in Browser Engines



13th of February, 2024

PhD defense of Gertjan Franken