



Web Platform Threats: Automated Detection of Web Security Issues With WPT



Pedro Bernardo¹



@bl4ck_pwn



pedro.bernardo@tuwien.ac.at

Shared first
authorship with:

Lorenzo Veronese¹



@310wert



lorenzo.veronese@tuwien.ac.at

With:

Valentino **Dalla Valle**², Stefano **Calzavara**²
Marco **Squarcina**¹, Pedro **Adão**³, Matteo **Maffei**¹



¹TU Wien ² Università Ca' Foscari Venezia ³ IST, Universidade de Lisboa

August 14-16, 2024 // Philadelphia, PA, USA

The Web Platform

Specifications

Web Platform Tests

Experts



manual reviews



Content Security Policy Level

Fetch Living Standard — Last Updated 24

Service Workers Nightly

Workgroup: HTTP
Internet-Draft: draft-ietf-httpbis-rtc6265bis-11
Obsoletes: 5100 (if approved)
Published: 7 November 2022
Intended Status: Standards Track
Expires: 11 May 2023

5. Bingler, Ed., Google LLC
M. West, Ed., Google LLC
J. Willander, Ed., Apple, Inc.

Abstract

This document defines the HTTP Cookie and Set-Cookie header fields. These header fields can be used by HTTP servers to store state (called cookies) at HTTP user agents, letting the servers maintain a stateful session over the mostly stateless HTTP protocol. Although cookies have many historical infelicities that degrade their security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet. This document obsoletes RFC 6265.

About This Document

This note is to be removed before publishing as an RFC.

Status Information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-httpbis-rtc6265bis/>.

Discussion of this document takes place on the HTTP Working Group mailing list (public-ietf-httpbis-wg@googlegroups.com), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/>. Working Group information can be found at <https://httpwg.org/>.

Source for this draft and an issue tracker can be found at <https://github.com/ietf-wg-httpbis-rtc6265bis>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 May 2023.



compliance tests



Web Browsers



implementations



The Web Platform

Specifications

Content Security Policy Level W3C

Fetch
Living Standard — Last Updated 24

Service Workers Nightly W3C
Editor's Draft, 11 April 2023

Workgroup: HTTP
Internet-Draft: Google LLC
draft-ietf-httpbis-rfc6265bis-11
Obsoletes: 6265 (If Approved)
Published: 7 November 2022
Intended Status: Standards Track
Expires: 11 May 2023

S. Bingler, Ed.
M. West, Ed.
Google LLC
J. Wilander, Ed.
Apple, Inc

Datatracker
draft-ietf-httpbis-rfc6265bis-11
Internet-Draft

Info Contents

Document type
Active Internet-Draft (WG)

Select version

00	01	02	03	04
07	08	09	10	11

Compare versions

draft-ietf-httpbis-rfc6265

draft-ietf-httpbis-rfc6265

side-by-side Inline

Authors
[Steven Bingler](#), [Mike John Wilander](#)
[Email authors](#)

Replaces
draft-ietf-httpbis-cookies
draft-thomson-http-om
draft-ietf-httpbis-cookiesite
draft-ietf-httpbis-cookiesite

Abstract

This document defines the HTTP Cookie and Set-Cookie header fields. These header fields can be used by HTTP servers to store state (called cookies) at HTTP user agents, letting the servers maintain a stateful session over the mostly stateless HTTP protocol. Although cookies have many historical infelicities that degrade their security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet. This document obsoletes RFC 6265.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-httpbis-rfc6265bis/>.

Discussion of this document takes place on the HTTP Working Group mailing list (<mailto:ietf-http-wg@4.org>), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/>. Working Group information can be found at <https://httpwg.org/>.

Source for this draft and an issue tracker can be found at <https://github.com/httpwg/http-extensions/labels/6265bis>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

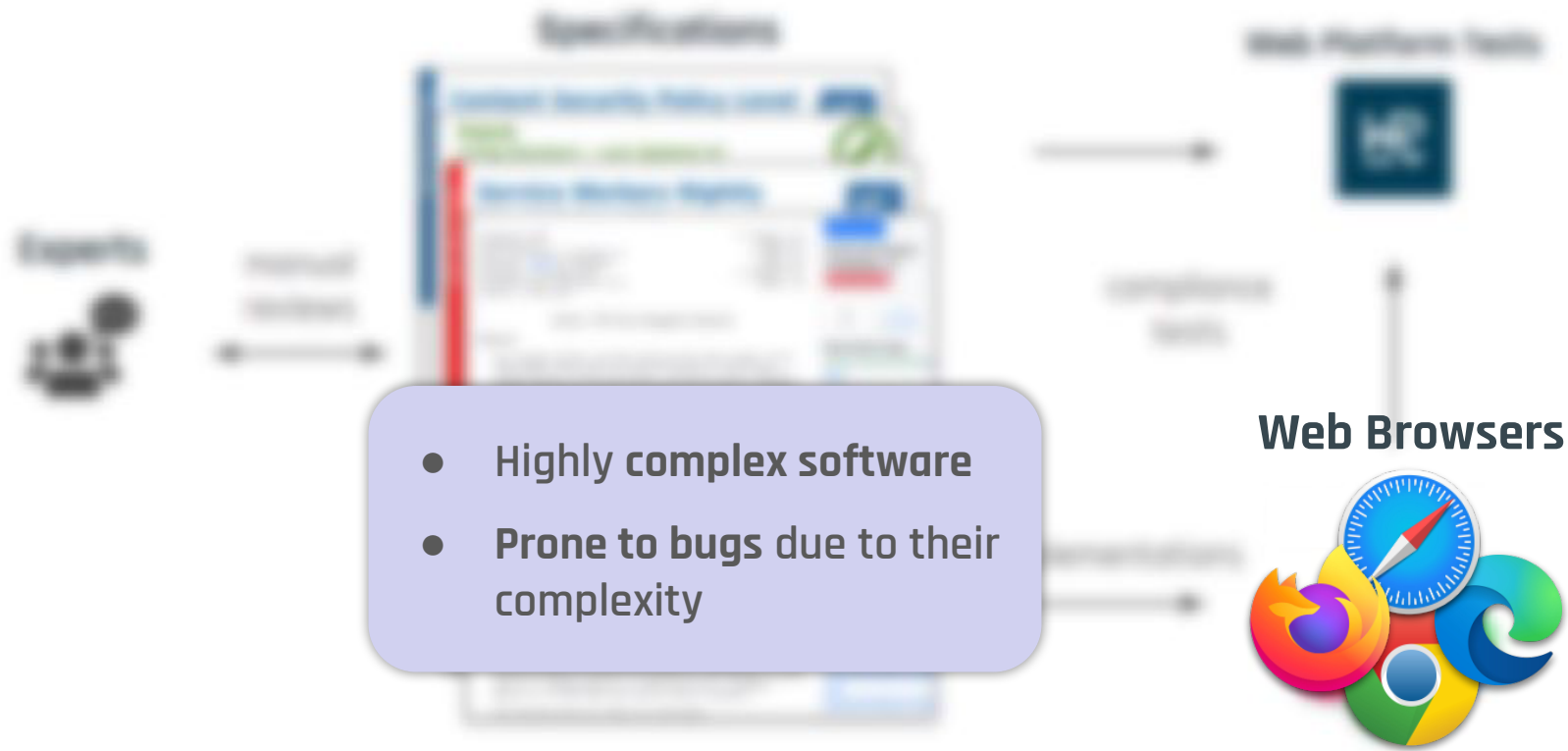
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 May 2023.

- Written informally in natural language
- Unintended interactions lead to security issues

The Web Platform



The Web Platform

web-platform-tests dashboard

Latest Run Recent Runs Interop 2024 Insights Processor About

wpt

Search test files, like 'cors/allow-headers.htm', then press <Enter>

For information on the search syntax, [view the search documentation](#)

Browser Specific Failures graph (click the arrow to expand)

Showing 56233 tests (1873930 subtests) from the latest master test runs for chrome[stable], firefox[stable], safari[stable]

Path	Chrome 126 Linux 20.04 1054674 Jul 19, 2024	Firefox 128 Linux 20.04 1054674 Jul 19, 2024	Safari 17.5 macOS 14.5 1054674 Jul 19, 2024
accelerometer/	141 / 150	8 / 150	8 / 150
accessibility/	62 / 62	62 / 62	62 / 62
accname/	373 / 379	377 / 379	324 / 379
acid/	101 / 102	100 / 102	101 / 102
ambient-light/	47 / 50	4 / 50	4 / 50
animation-stylesheet/	18 / 18	18 / 18	18 / 18

Web Platform Test



Does not reason about security

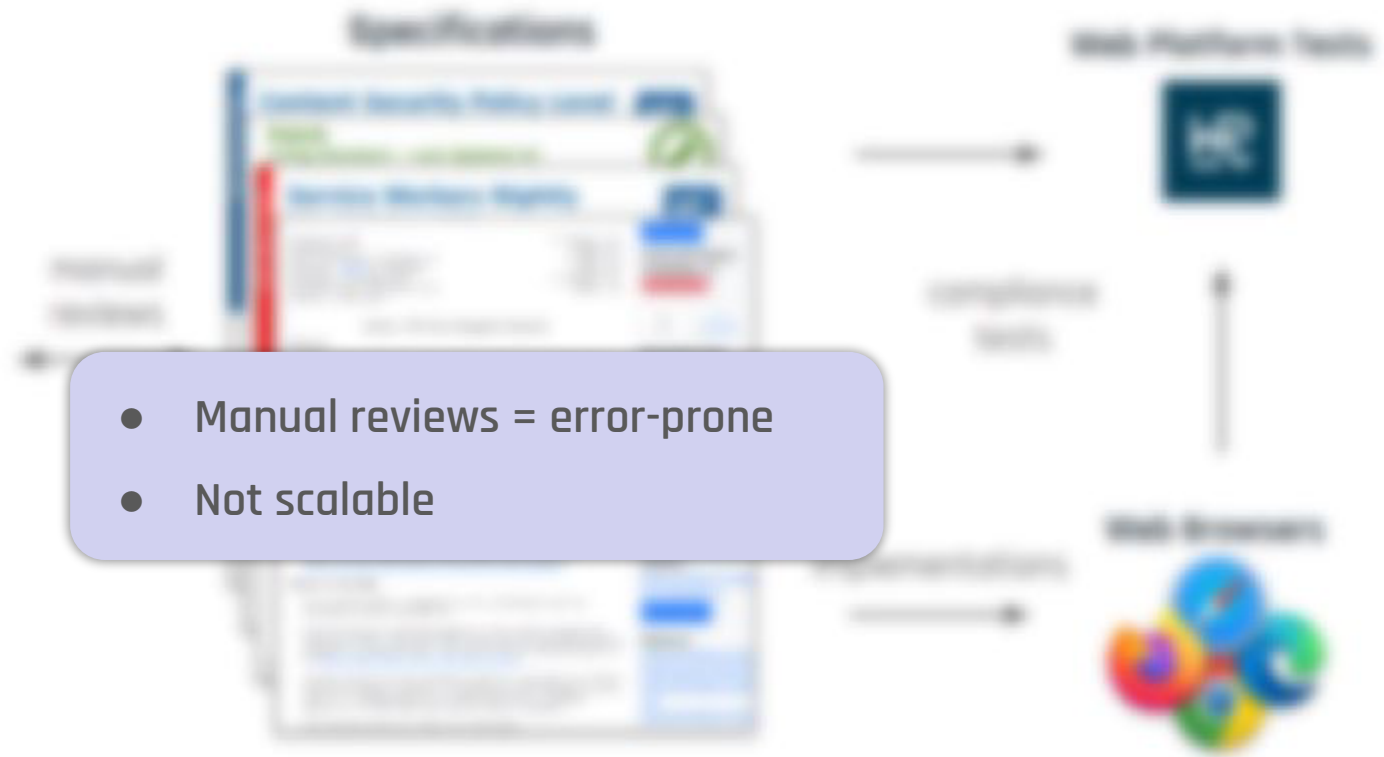
x-frame-options/	153 / 153
xhr/	2073 / 2203
Subtest Total	1829454 / 1898205

The Web Platform

Experts



- Manual reviews = error-prone
- Not scalable



Can we automate security testing?



Experts

manual reviews



Specifications

Content Security Policy Level W3C

Fetch Living Standard — Last Updated 24

Service Workers Nightly W3C

Workgroup: HTTP
Internet-Draft: draft-ietf-httpbis-rtc6265bis-11
Obsoletes: 520 (if approved)
Published: 7 November 2022
Intended Status: Standards Track
Expires: 11 May 2023

5. Bingle, Ed., Google LLC
M. West, Ed., Google LLC
J. Willander, Ed., Apple, Inc

draft-ietf-httpbis-rtc6265bis-11
Internet-Draft

Info Contents

Document type
Active Internet-Draft (WG)

Select version
00 01 02 03 04
07 08 09 10 11

Compare versions
draft-ietf-httpbis-rtc6265
draft-ietf-httpbis-rtc6265bis-11

Side by side inline

Authors
Steven Bingle, Ed., Mike West, Ed., John Willander, Ed.

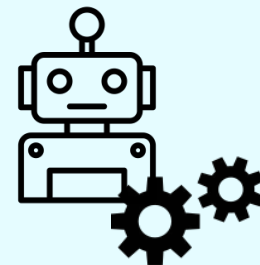
Replaces
draft-ietf-httpbis-rtc6265bis-10
draft-thomson-http-om
draft-ietf-httpbis-cookies
site
draft-ietf-httpbis-cookies

Abstract
This document defines the HTTP Cookie and Set-Cookie header fields. These header fields can be used by HTTP servers to store state (called cookies) at HTTP user agents, letting the servers maintain a stateful session over the mostly stateless HTTP protocol. Although cookies have many historical infelicities that degrade their security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet. This document obsoletes RFC 6265.

About This Document
This note is to be removed before publishing as an RFC.
Status Information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-httpbis-rtc6265bis/>.
Discussion of this document takes place on the HTTP Working Group mailing list (<mailto:ietf-httpbis@ietf.org>), which is archived at <https://lists.w3.org/Archives/Public/ietf-httpbis-wg/>. Working Group information can be found at <https://w3ctag.org/>.
Source for this draft and an issue tracker can be found at <https://github.com/stevenbingle/rtc6265bis>.

Status of This Memo
This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on 11 May 2023.

Web Platform Tests



compliance tests



Web Browsers



implementations

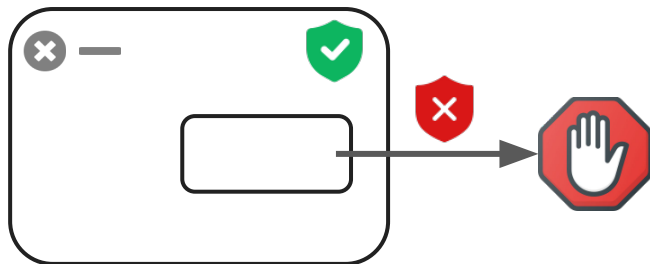


Web Invariants



Intended security properties of the Web that are always expected to hold

Web Invariant - Blockable Mixed Content Filtering



Insecure (http) resources fetched from **secure** (https) pages should be **blocked**

Web Invariants



Intended security properties of the Web that are always expected to hold

IEEE S&P 2023

WebSpec: Towards Machine-Checked Analysis of Browser Security Mechanisms

Lorenzo Veronese*, Benjamin Farinier[†], Pedro Bernardo*, Mauro Tempesta*, Marco Squarcina*, Matteo Maffei[†]
*TU Wien
[†]Univ Rennes, Inria, CNRS, IRISA

The screenshot shows two overlapping W3C Working Draft pages. The top page is for 'Content Security Policy Level 3' (W3C Working Draft, 11 April 2023) and the bottom page is for 'Service Workers Nightly' (Editor's Draft, 11 April 2023). Both pages include metadata such as 'Latest published', 'Editors', 'Participate', and 'Abstract'. The 'Service Workers Nightly' page also features a 'Document type' section set to 'Active Internet-Draft' and a 'Compare versions' section.

Specifications



Implementations

Web Platform Threats: Overview

Web Invariants



Cookies



Mixed Content

Web Platform Threats: Overview

Web Invariants



Cookies



Mixed Content



Instrumented
Browsers

Web Platform Threats: Overview

Web Invariants

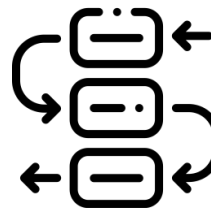


Cookies



Mixed Content

Execution Traces



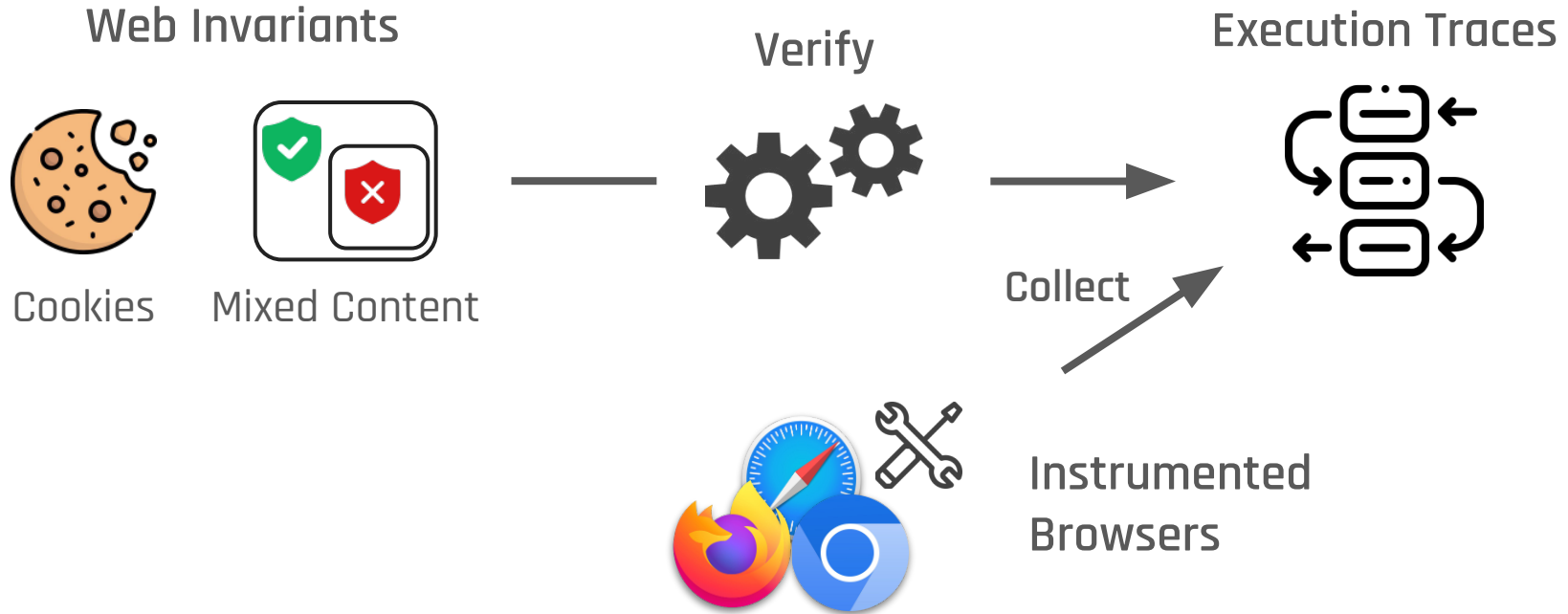
Collect



Instrumented
Browsers



Web Platform Threats: Overview



Web Platform Threats: Challenges

No clear algorithm for defining invariants

Web Invariants



Cookies

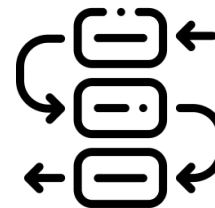


Mixed Content

Verify



Execution Traces

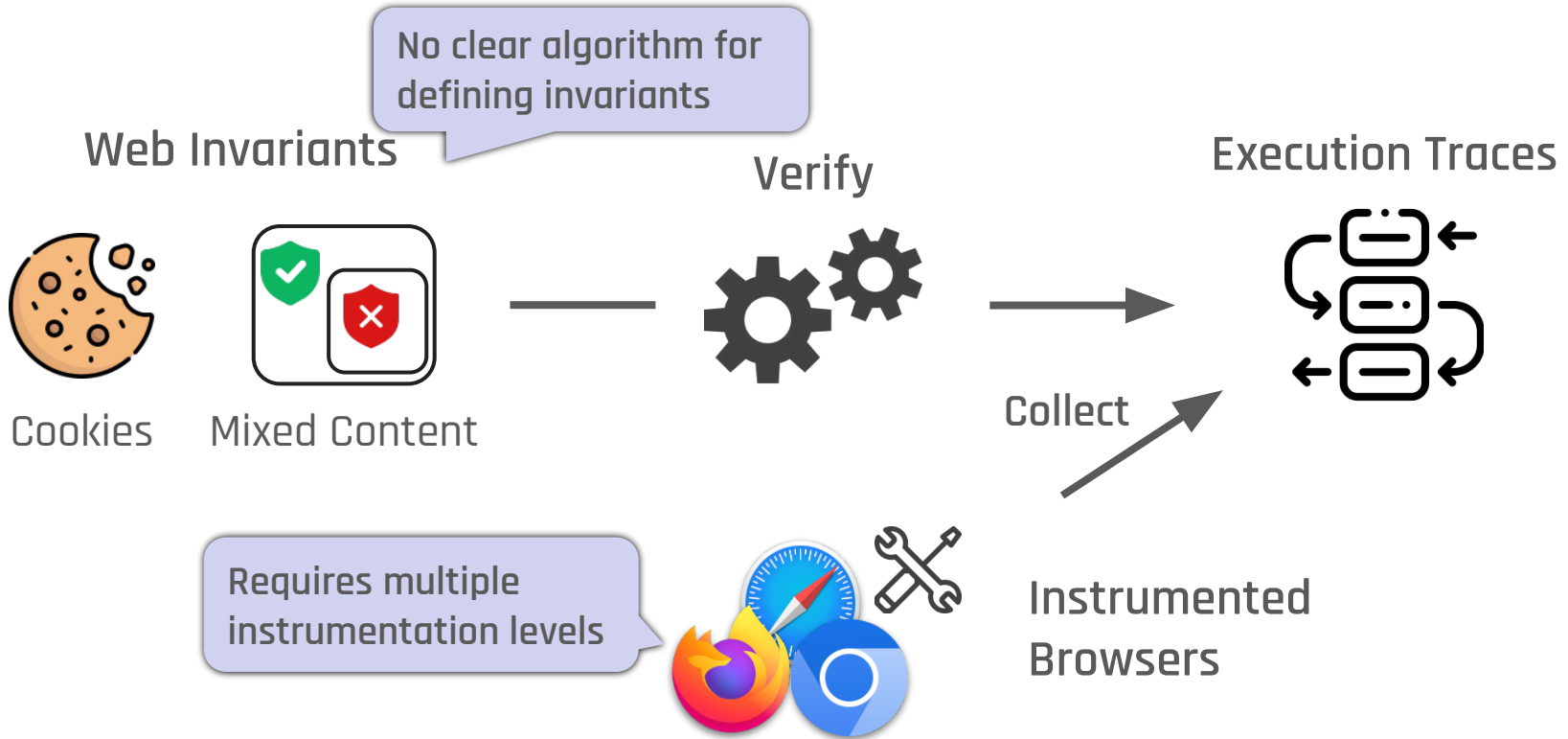


Collect

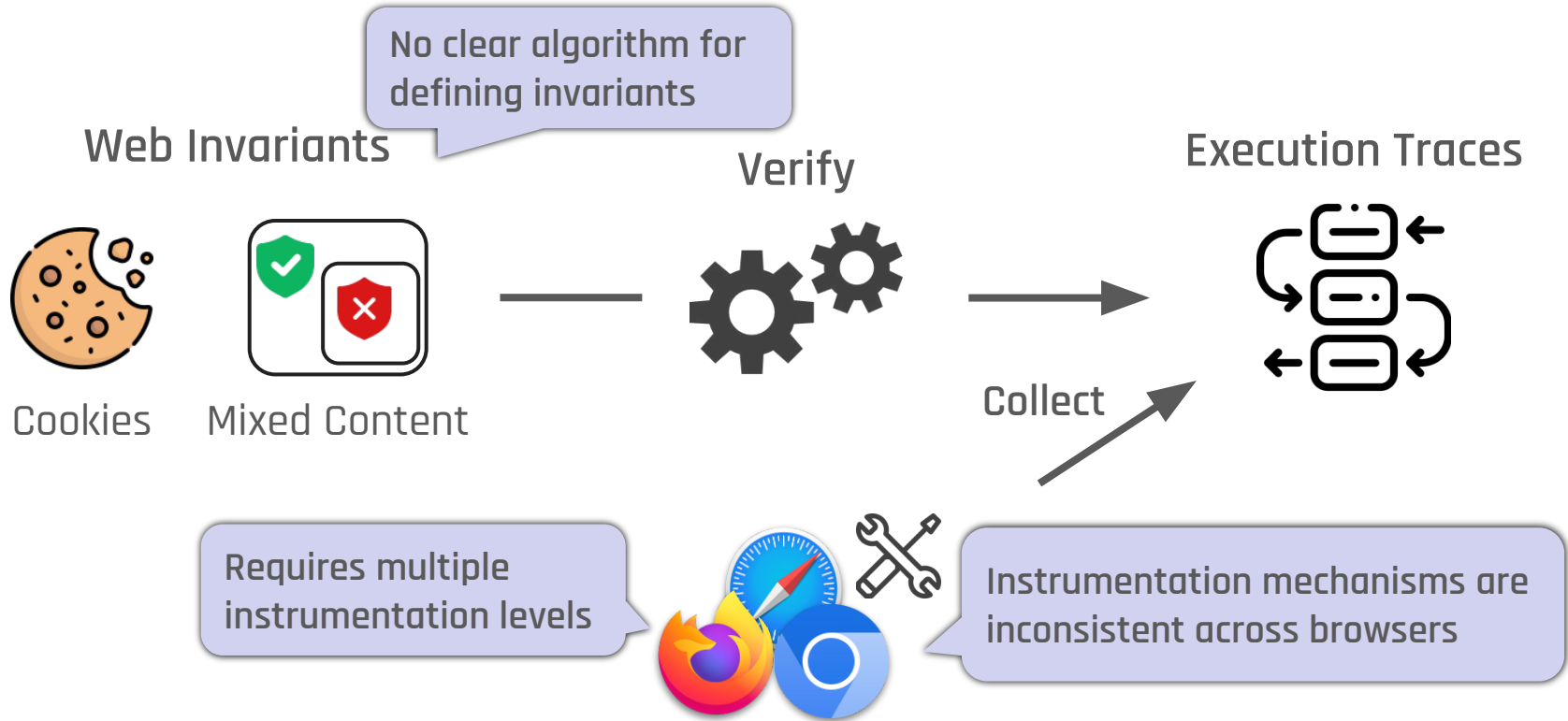
Instrumented
Browsers



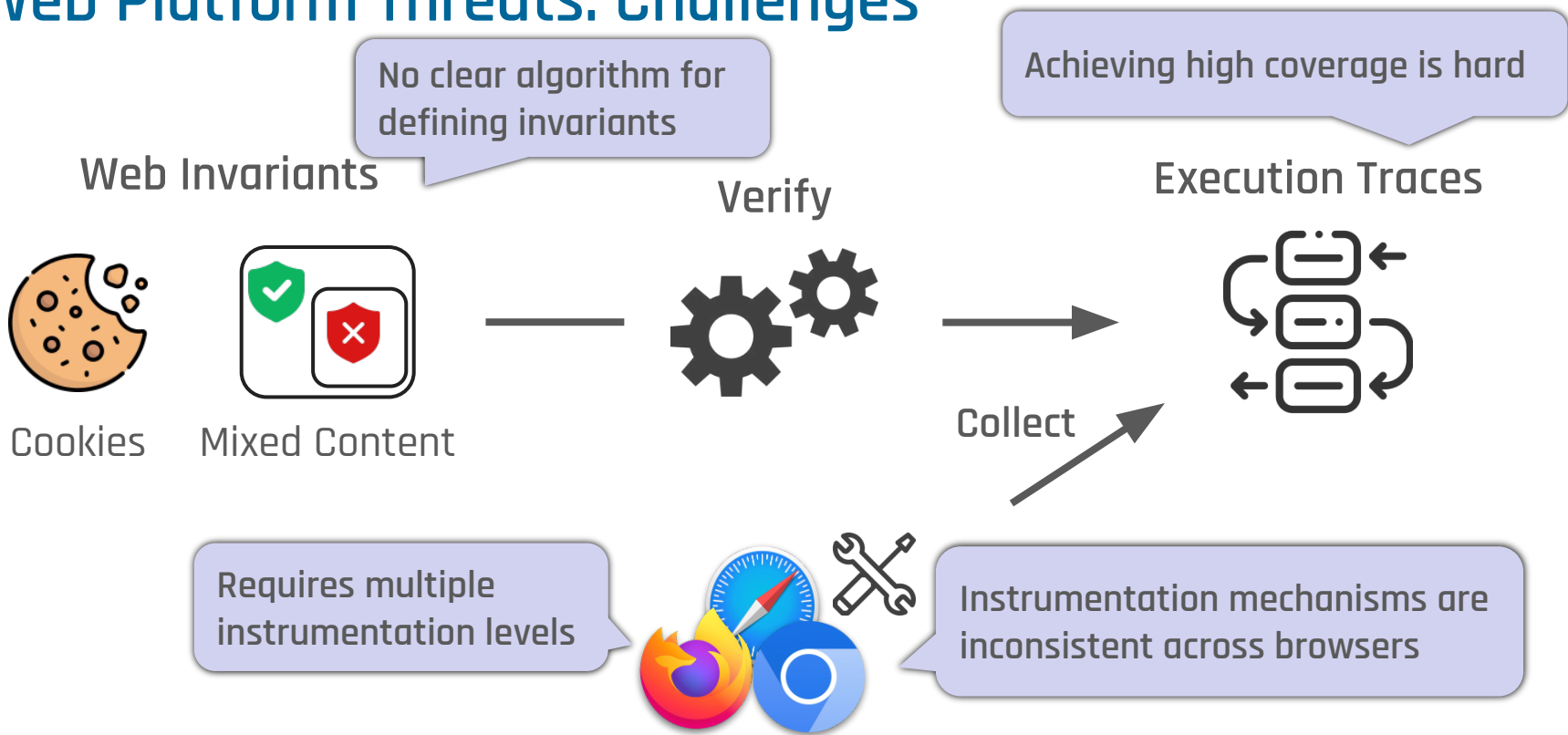
Web Platform Threats: Challenges



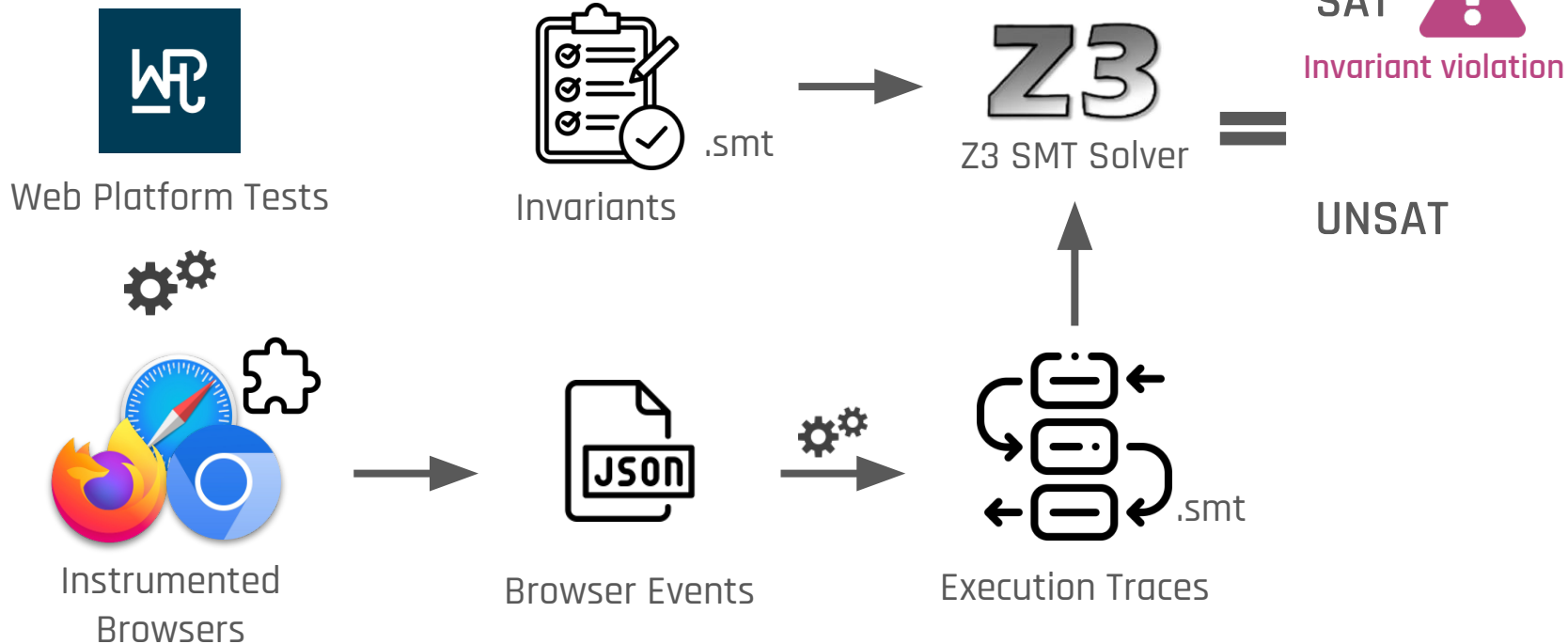
Web Platform Threats: Challenges



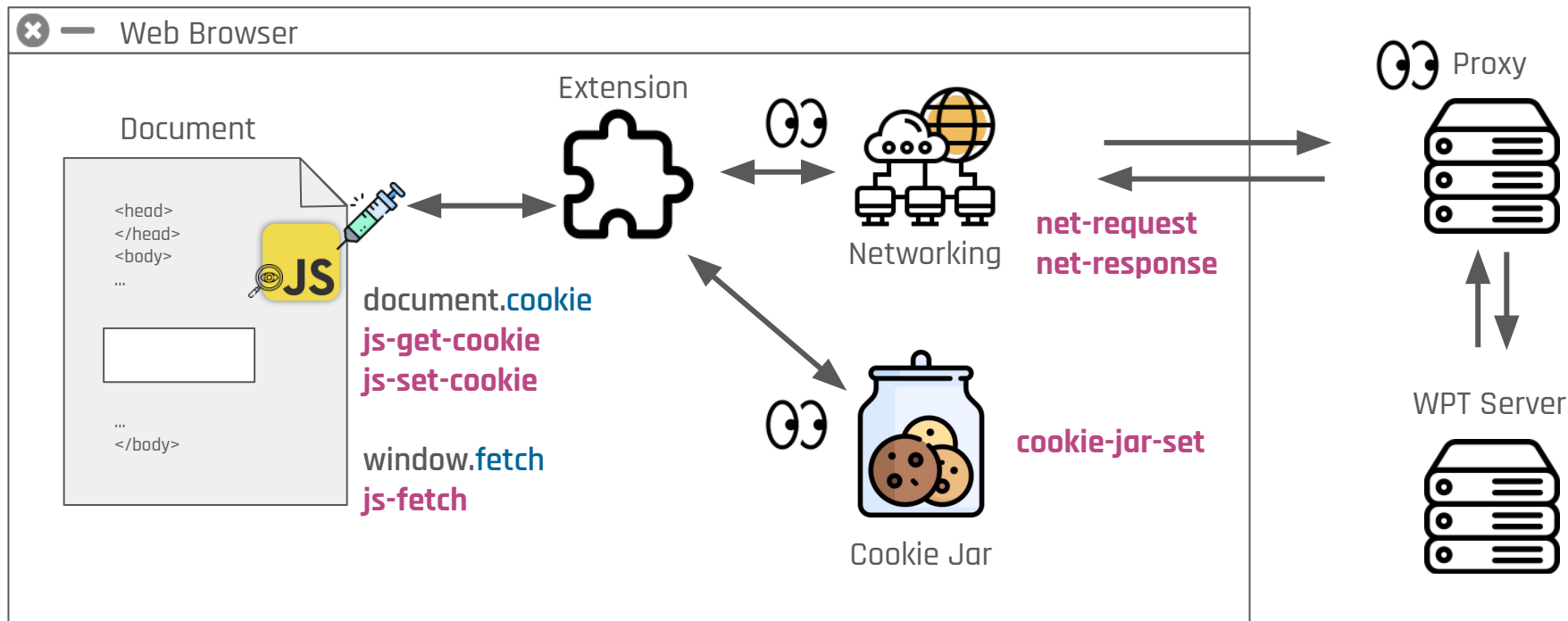
Web Platform Threats: Challenges



Web Platform Threats: Pipeline



Browser Instrumentation and Events



Execution Traces

Code

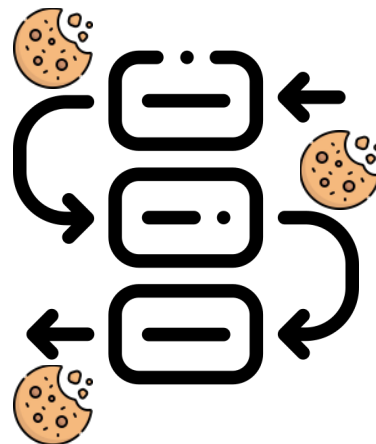
```
✕ — ✓ https://domain/path/test.html

test () {
  document.cookie = "A=B; secure; path=/";
  assert document.cookie.includes("A=B")
}
```

Execution Trace

```
t1@ js-set-cookie(Ctx, "A=B; secure; path=/")
t2@ cookie-jar-set("A", "B", secure=true, path="/", domain="domain")
t3@ js-get-cookie(Ctx, "A=B")
```

Ctx = < ID, "https://domain/path/test.html" >



Results



Cookies







Mixed Content

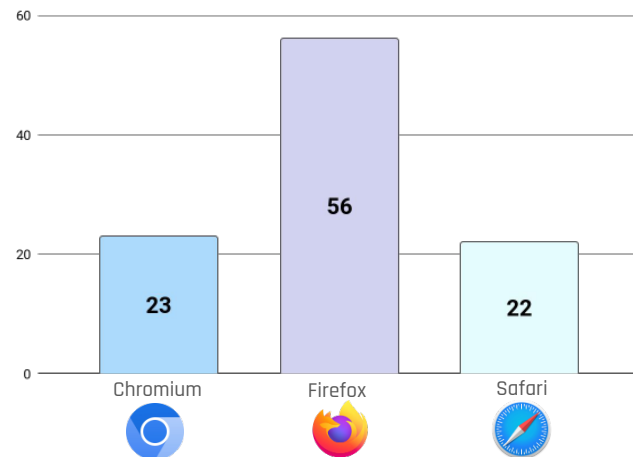
	Valid
I.1 Integrity of Secure cookies ^[1]	✓
I.2 Confidentiality of HttpOnly cookies ^[1]	✓
I.3 Integrity of __Host- Cookies ^[1]	✓
I.4 Integrity of SameSite cookies	✗
I.5 Isolation of SameSite cookies	✗
I.6 Cookie serialization collision resistance	✗
I.7 Confidentiality of Secure cookies	✓
I.8 Blockable mixed content filtering	✗
I.9 Upgradeable mixed content filtering	✗







[1] "WebSpec: Towards Machine-Checked Analysis of Browser Security Mechanisms" 2023 IEEE S&P L. Veronese, et. al

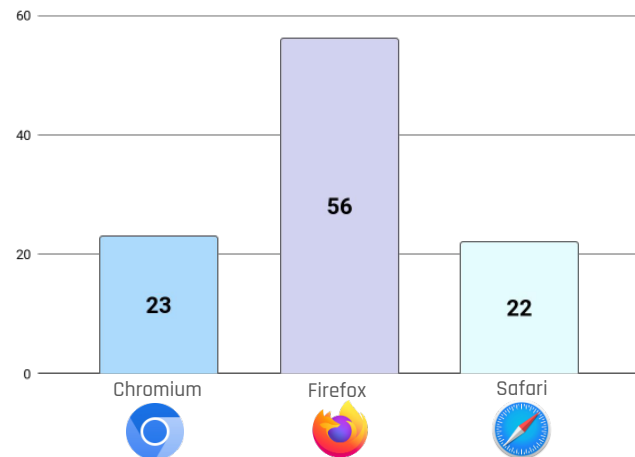
Results

- **101 Invariant violations** →
- **10 vulnerabilities** 
 - 8 individual reports
 -  **CVE-2023-38592**
Mixed Content Policy bypass via framing
 -  **CVE-2024-6611**
SameSite cookie integrity violation
-  **Changes to rfc6265bis**
 - Clarified security implications of SameSite cookies against XS-Leaks and CSRF
 - Include support for *potentially trustworthy origins* in "secure protocol" checks



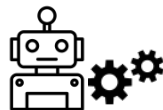
Results

- **101 Invariant violations** →
- **10 vulnerabilities** 
 - 8 individual reports
 -  **CVE-2023-38592**
Mixed Content Policy bypass via framing
 -  **CVE-2024-6611**
SameSite cookie integrity violation
-  **Changes to rfc6265bis**
 - Clarified security implications of SameSite cookies against XS-Leaks and CSRF
 - Include support for *potentially trustworthy origins* in "secure protocol" checks



- **15 false positive traces**
 - Missing events
 - Incorrectly ordered events
 - Missing information in events

Conclusion



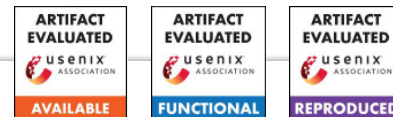
Automated analysis can detect invariant violations during development



Limitations can be mitigated with better instrumentation support



Defining Web invariants should be a priority



Web Platform Threats: Automated Detection of Web Security Issues With WPT

Pedro Bernardo[¶], Lorenzo Veronese[¶], Valentino Dalla Valle[‡],
Stefano Calzavara[‡], Marco Squarcina[‡], Pedro Adão[§], Matteo Maffei[†]

[†] TU Wien

[‡] Università Ca' Foscari Venezia

[§] Instituto Superior Técnico, Universidade de Lisboa, and Instituto de Telecomunicações

Abstract

Client-side security mechanisms implemented by Web browsers, such as cookie security attributes and the Mixed Content policy, are of paramount importance to protect Web applications. Unfortunately, the design and implementation of such mechanisms are complicated and error-prone, potentially exposing Web applications to security vulnerabilities. In this paper, we present a practical framework to formally and automatically detect security flaws in client-side security mechanisms. In particular, we leverage Web Platform Tests (WPT), a popular cross-browser test suite, to automatically collect browser execution traces and match them against Web invariants, i.e., intended security properties of Web mechanisms expressed in first-order logic. We demonstrate the effectiveness of our approach by validating 9 invariants against the WPT test suite, discovering violations with clear security implications in 104 tests for Firefox, Chromium and Safari. We disclosed the root causes of these violations to browser vendors and standard bodies, which resulted in 8 individual reports and one CVE on Safari.

1 Introduction

Writing secure Web applications is notoriously hard, due to the heterogeneity, complexity and open-ended nature of the Web. To mitigate the challenges of secure Web application development, browsers integrate a growing list of client-side security mechanisms to assist Web developers. Examples of such mechanisms include cookie security attributes (HttpOnly,

flaws, which led to breaking well-established Web security invariants [13, 33]. *Formal methods proved to be an essential tool to rigorously analyze client-side security mechanisms, allowing for the identification of bugs and formulation of formal security proofs in such a complex environment.* All state-of-the-art techniques, however, be they manual [20], machine-checked [17], or automated [13, 33], apply to *browser models*, which suffer from two fundamental drawbacks. First, client-side security mechanisms evolve over time and new ones are being proposed on a regular basis, which makes browser models extremely hard to maintain. “are correct, security-critical bug tions [23, 29, 30, 36]. Correctly mechanisms within browsers is for various reasons. Browsers are ware artifacts: for instance, the roughly 35 million lines of code. kernel. Furthermore, browser ve natural language specifications Web Consortium (W3C), into n already complicated codebase. rity mechanisms often cannot be them interact with core browser defines requests, responses, and binds them. This means that t side security mechanisms often browser components which we integration in mind.

We thus tackle the following *design a practical framework to*



Web Platform Threats: Automated Detection of Web Security Issues With WPT

Thank you!

Pedro Bernardo

 @bl4ck_pwn

 pedro.bernardo@tuwien.ac.at

Icons by **flaticon**

