BUIzz:

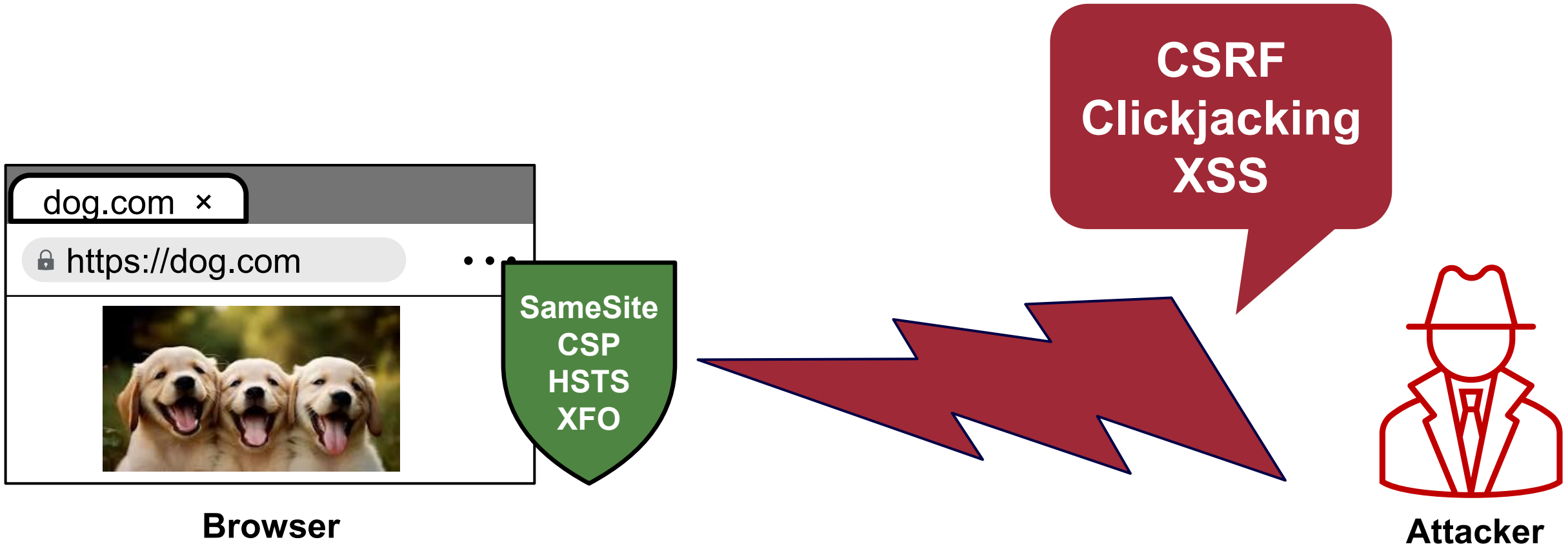
Finding Policy Enforcement Bugs via Interaction Simulation on the Browser User Interface

USENIX Security ’26

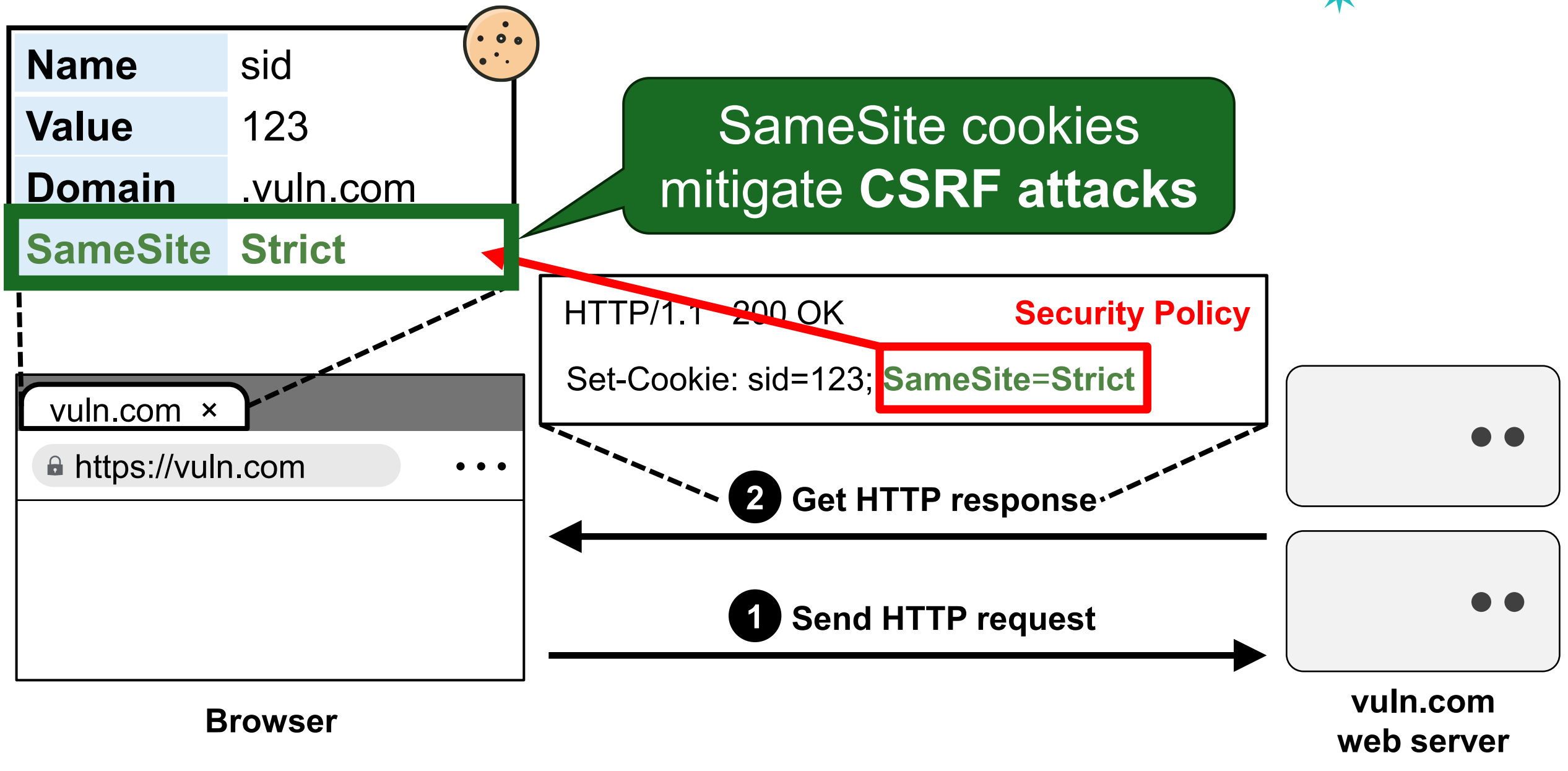
Mingi Jung, Donggyu Kim, Mijung Kim, Seongil Wi

UNIST

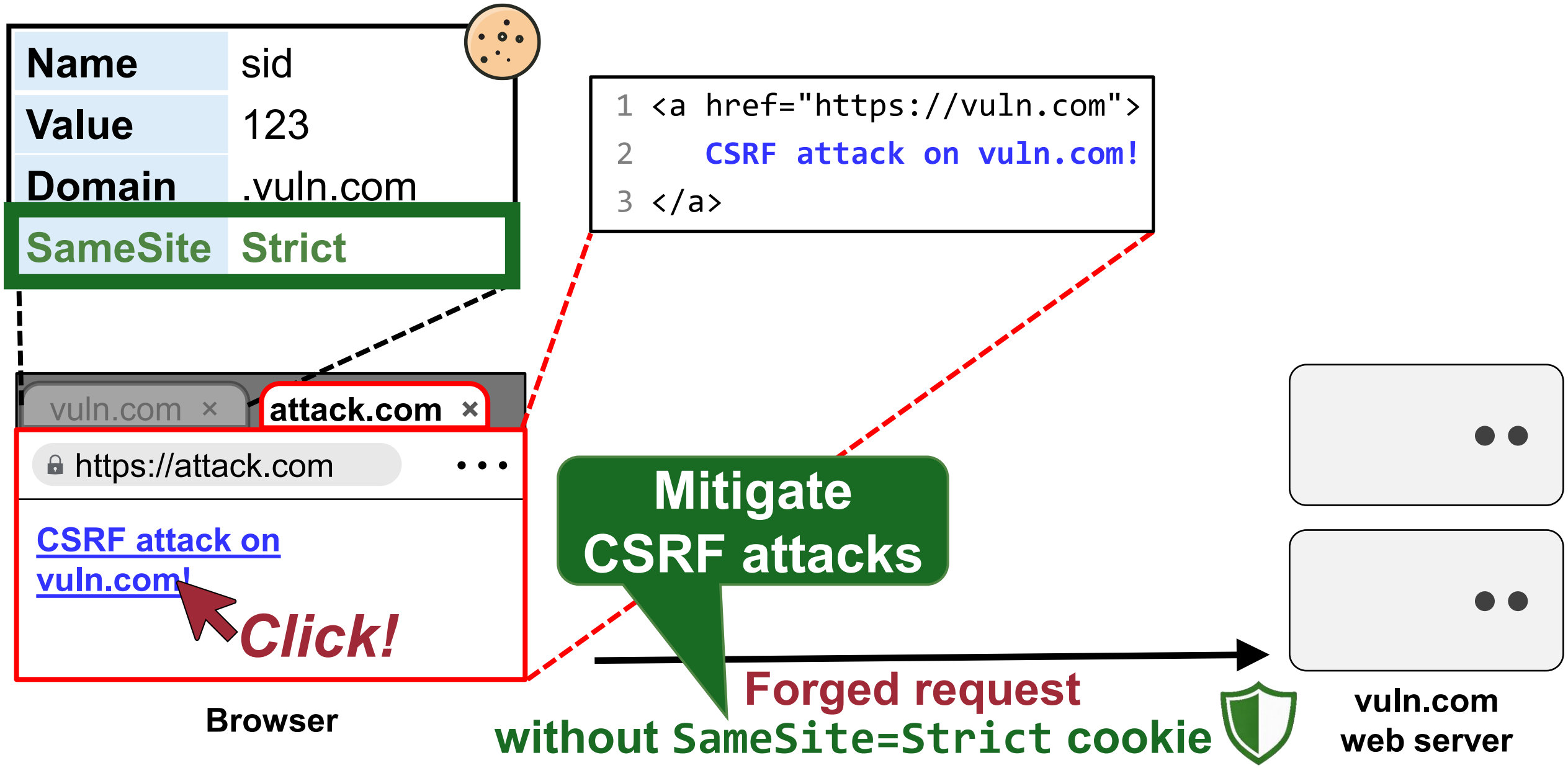
Client-side Security Policy



Policy Example: SameSite Cookie



SameSite Cookies Mitigate CSRF Attacks



Policy Enforcement Bugs

But what happens **if a browser <u>incorrectly enforces</u>** a security policy?

→ Fail the defense (reopen security holes)

Browser

Policy Enforcement Bugs

Forged request without SameSite=Strict cookie

vuln.com web server

Policy Enforcement Bugs

But what happens **if a browser <u>incorrectly enforces</u>** a security policy?

→ Fail the defense (reopen security holes)

Browser

- SameSite cookie bypass → reopen **CSRF attacks**
- CSP bypass → reopen **XSS attacks**
- HSTS bypass → reopen **MitM attacks**
- COOP bypass → reopen **XS-leak attacks**
- …

Forged request without SameSite=Strict cookie

vuln.com web server

Finding Policy Enforcement Bugs

Previous Studies

- DiffCSP, ***NDSS 2023***
- Who Left Open the Cookie Jar?, ***USENIX Security 2018***
- A Tale of Two Headers, ***USENIX Security 2020***
- Head(er)s Up!, ***CCS 2025***

Browser

Main limitation: *simulates only document-level interactions (i.e., click events)*

Missing attack surface: browser-UI level interactions

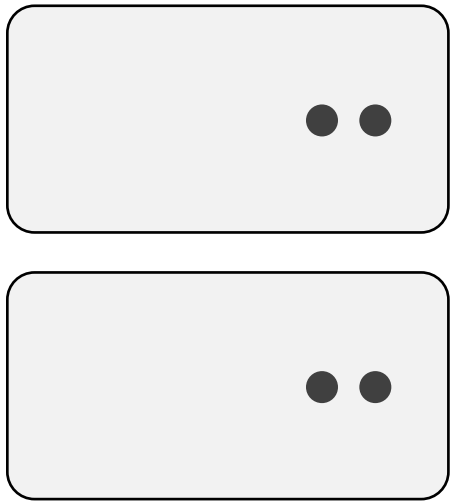
Motivating Example: CVE-2025-48980

Name	sid
Value	123
Domain	.victim.com
SameSite	**Strict**

Browser UI-level interaction:
Clicking “Open link in split view”
→ **SameSite cookie bypass**

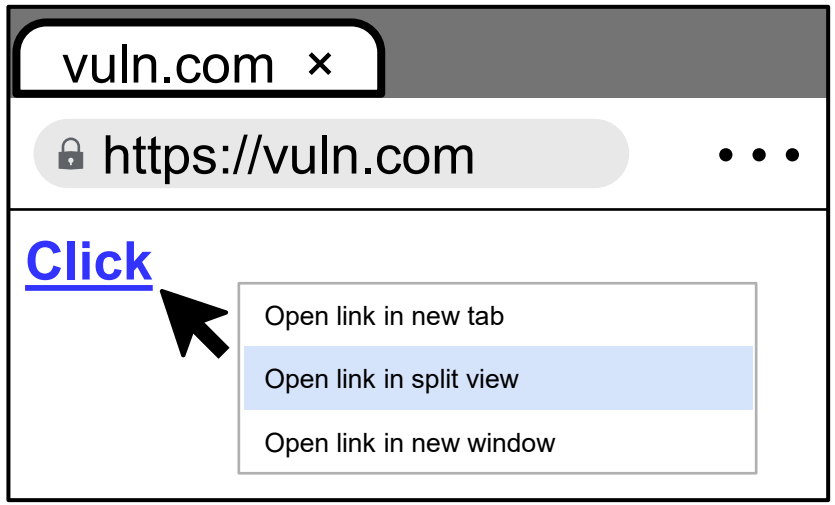
Click!

Forged request with `SameSite=Strict` cookie

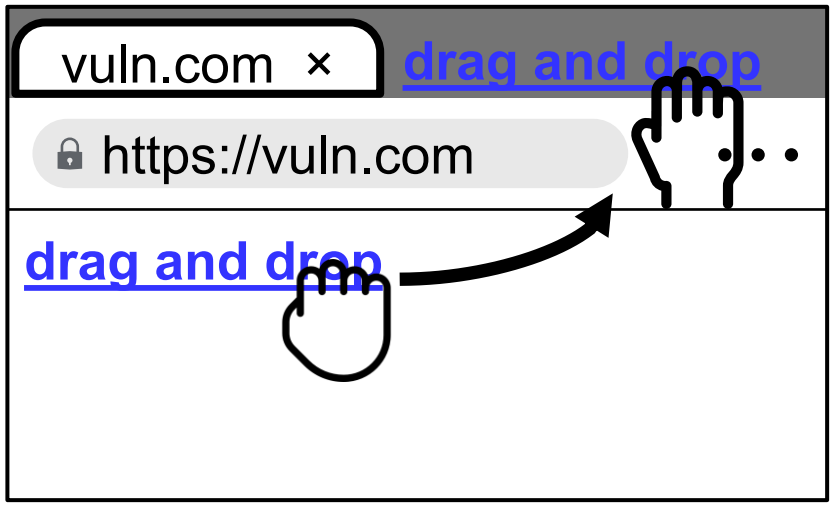


Our Goal:

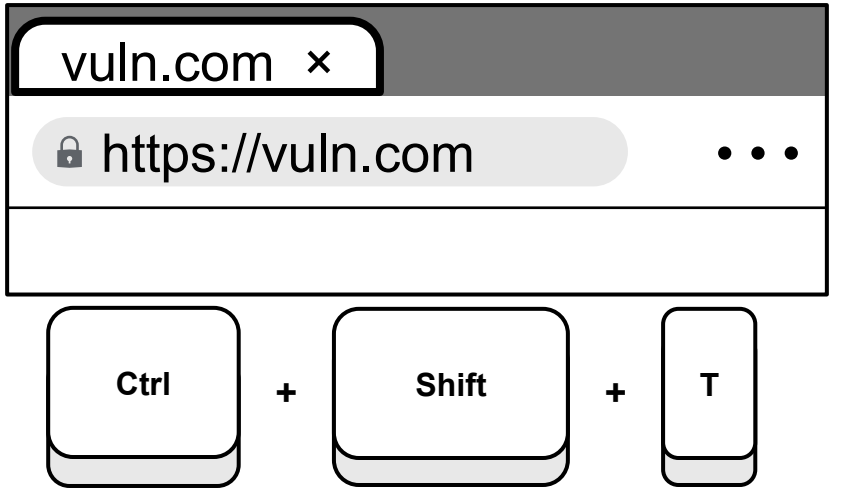
Finding policy enforcement bugs triggered by browser UI-level interactions



UI Interaction



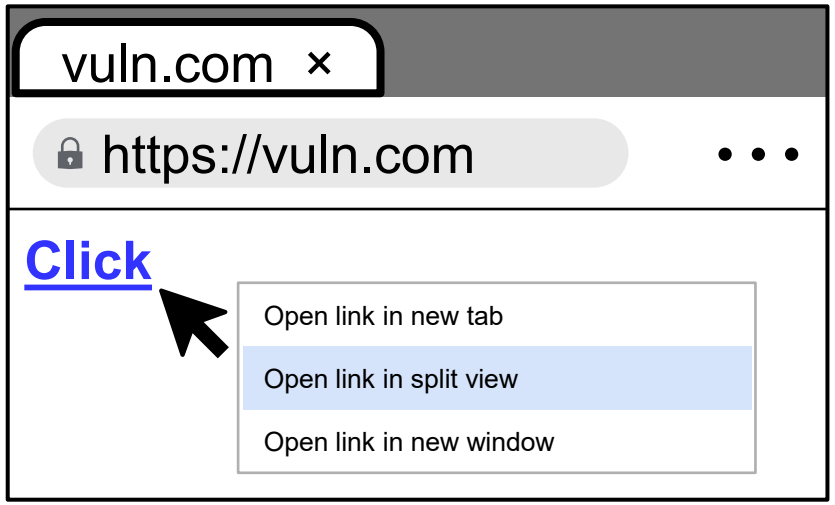
Mouse Interaction



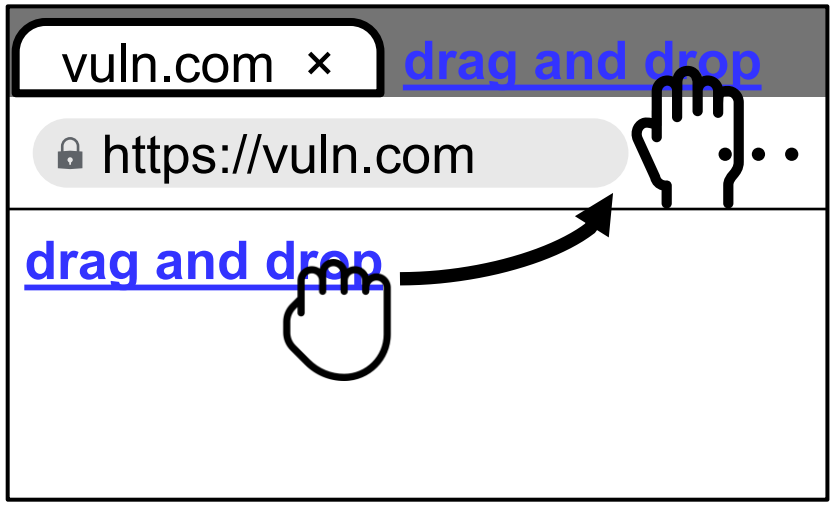
Keyboard Interaction

Challenges in Finding Bugs

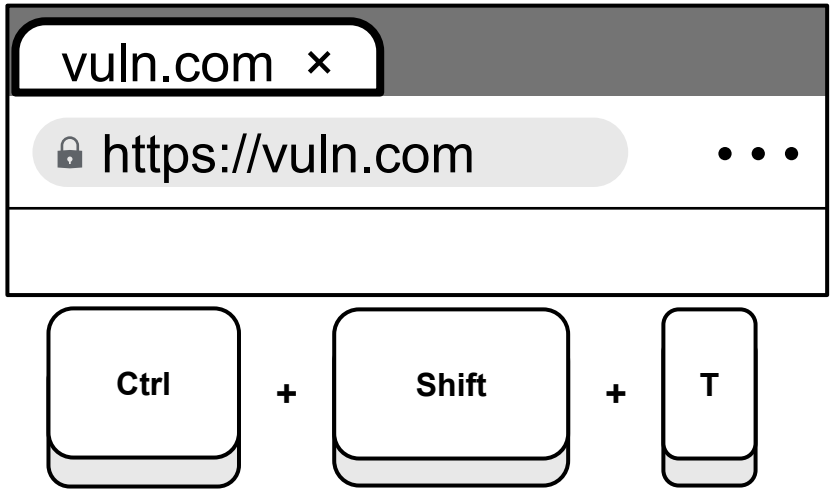
Large search space



UI Interaction



Mouse Interaction



Keyboard Interaction

Challenges in Finding Bugs

Large search space

Browser manuals

Action	Shortcut
Open a new window	**Ctrl + n**
Open a new window in Incognito mode	**Ctrl + Shift + n**
Open a new tab, and jump to it	**Ctrl + t**
Reopen previously closed tabs in the order they were closed	**Ctrl + Shift + t**
Jump to the next open tab	**Ctrl + Tab** or **Ctrl +**
Jump to the previous open tab	**Ctrl + Shift + Tab** or
Jump to a specific tab	**Ctrl + 1** through **Ctrl + 8**
Jump to the rightmost tab	**Ctrl + 9**
Open your home page in the current tab	**Alt + Home**
Open the previous page from your browsing history in the current tab	**Alt + Left arrow**
Open the next page from your browsing history in the current tab	**Alt + Right arrow**
Close the current tab	**Ctrl + w** or **Ctrl + F4**
Close the current window	**Ctrl + Shift + w** or **Alt + F4**
Minimize the current window	**Alt + Space** then **n**
Maximize the current window	**Alt + Space** then **x**
Quit Google Chrome	**Alt + f** then **x**
Move tabs right or left	**Ctrl + Shift + PgUp** or **Ctrl + Shift + PgDn**
Turn on full-screen mode	**F11**
Turn off full-screen mode	**F11** or press and hold **Esc**
Select multiple tabs	**F6** twice then **Shift + Ctrl + h**

Total: 486 interactions

Too many interactions create a vast search space

Challenges in Finding Bugs

Large search space

User interaction simulation

playwright

selenium

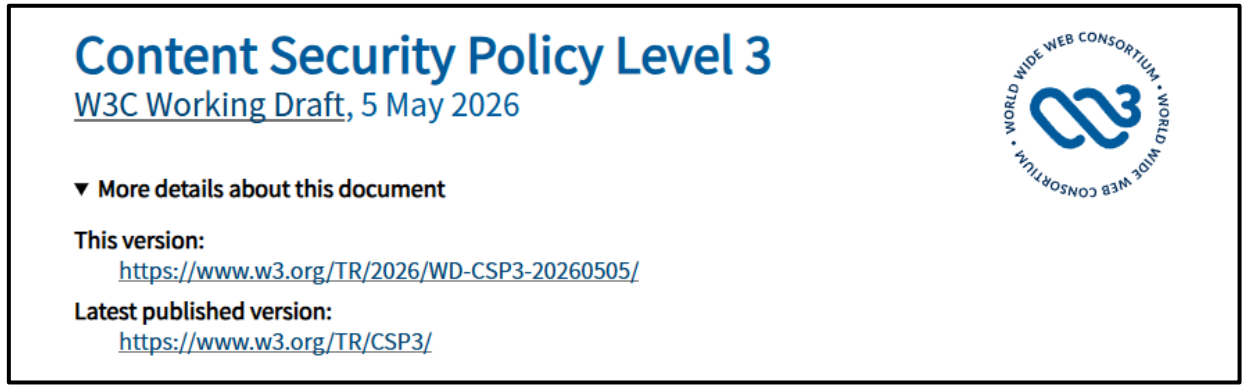
Cannot simulate browser UI-level interactions

Challenges in Finding Bugs

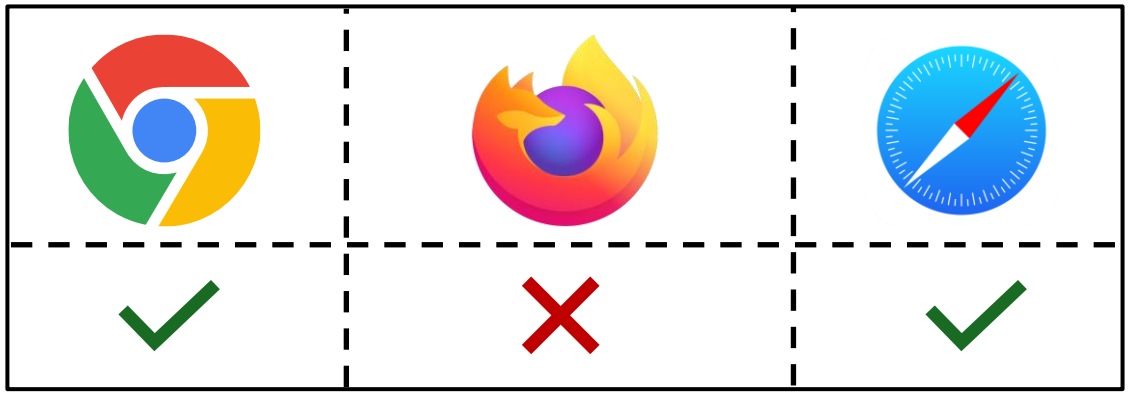
Large search space

User interaction simulation

Bug oracles implementation



Content Security Policy Spec



Existing solution:
Cross-browser differential testing

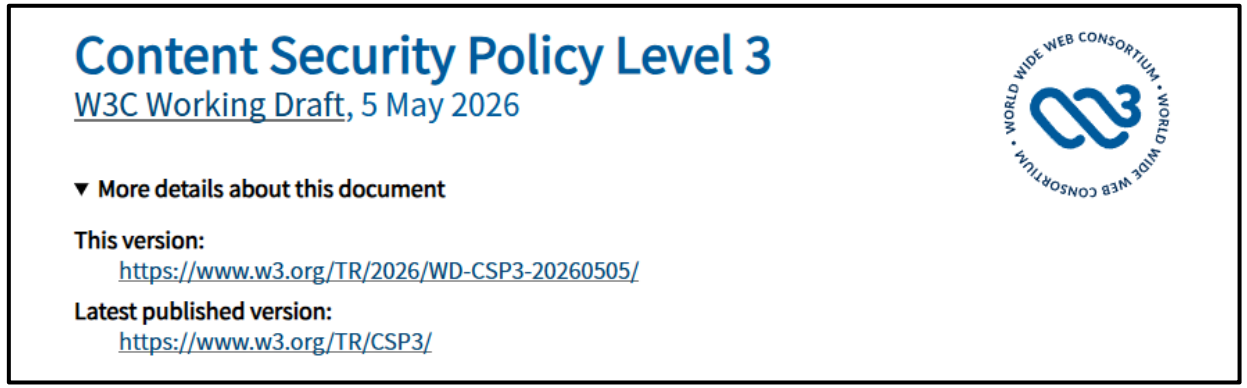
Complex specifications

Challenges in Finding Bugs

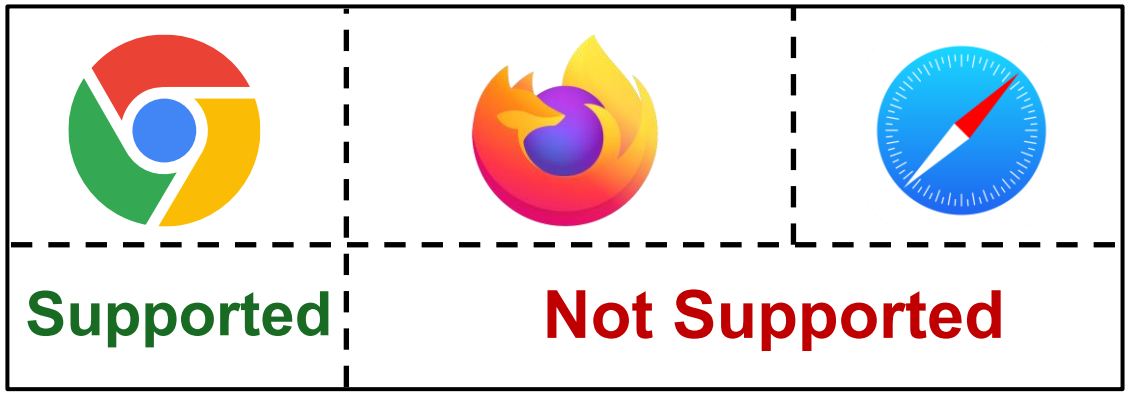
Large search space

User interaction simulation

Bug oracles implementation



Content Security Policy Spec



Existing solution:
Cross-browser differential testing

1. Time-consuming
2. Missing bugs common to all browsers
3. Inconsistent browser support (e.g. Permission Policy)

How we address all the challenges?

We propose

BUlzz

Our Approach

Large search space

User interaction simulation

Bug oracles implementation

Filtering for Navigation-related Interactions

Large search space

User interaction simulation

Bug oracles implementation

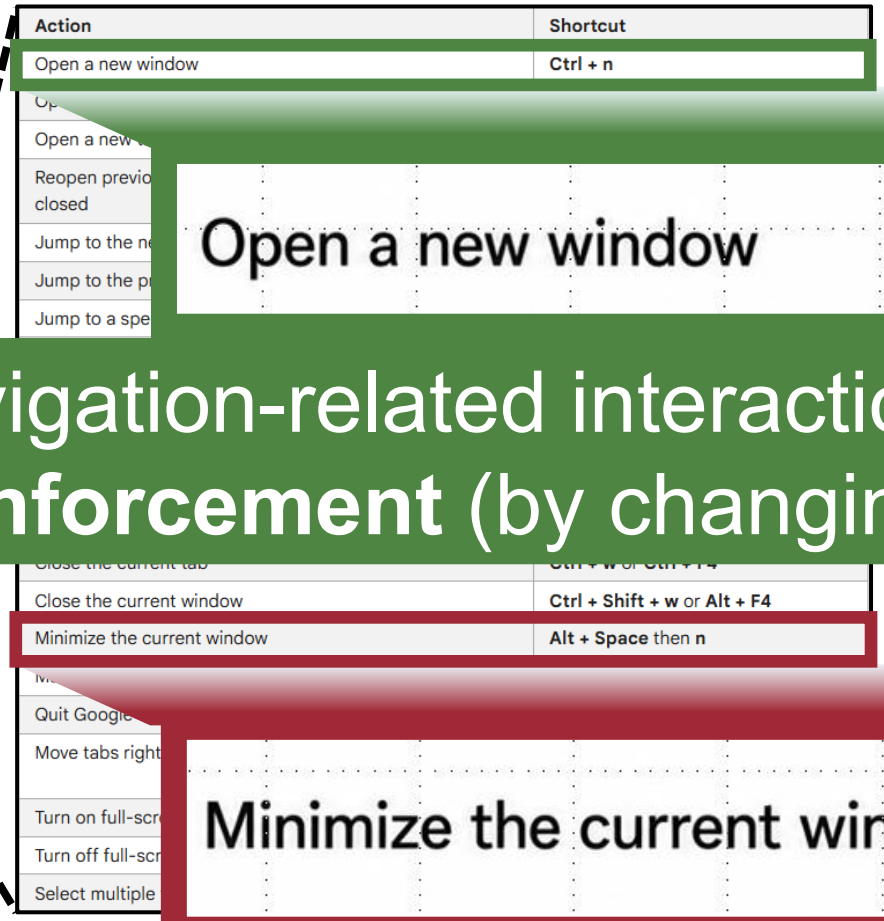
Filtering for navigation-related interactions

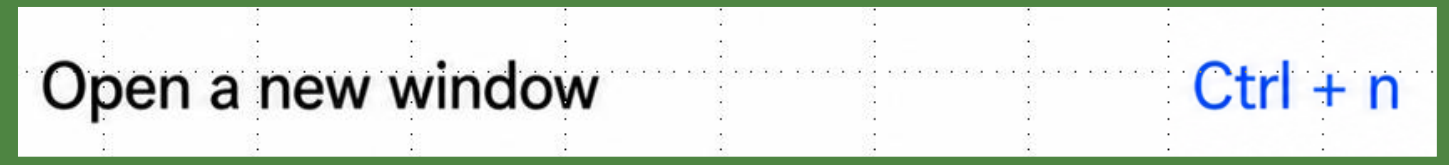
Filtering for Navigation-related Interactions

Large search space

Bug oracles

Browser manuals





Navigation-related interactions
→ **Affect policy enforcement** (by changing rendering state)

Non-navigation interactions
→ **Cannot affect policy enforcement**

Filtering for Navigation-related Interactions

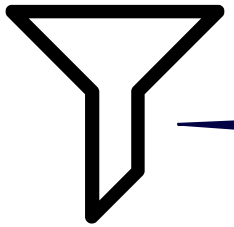
Large search space

User interaction simulation

Bug oracles implementation

Browser manuals

1. Collects all interactions



2. Keyword-based filtering (e.g., open)

Filtering for navigation-related interactions

Simulating at the OS Level

Large search space

User interaction simulation

Bug oracles implementation

Filtering for navigation-related interactions

Simulating at the OS Level

Large search space

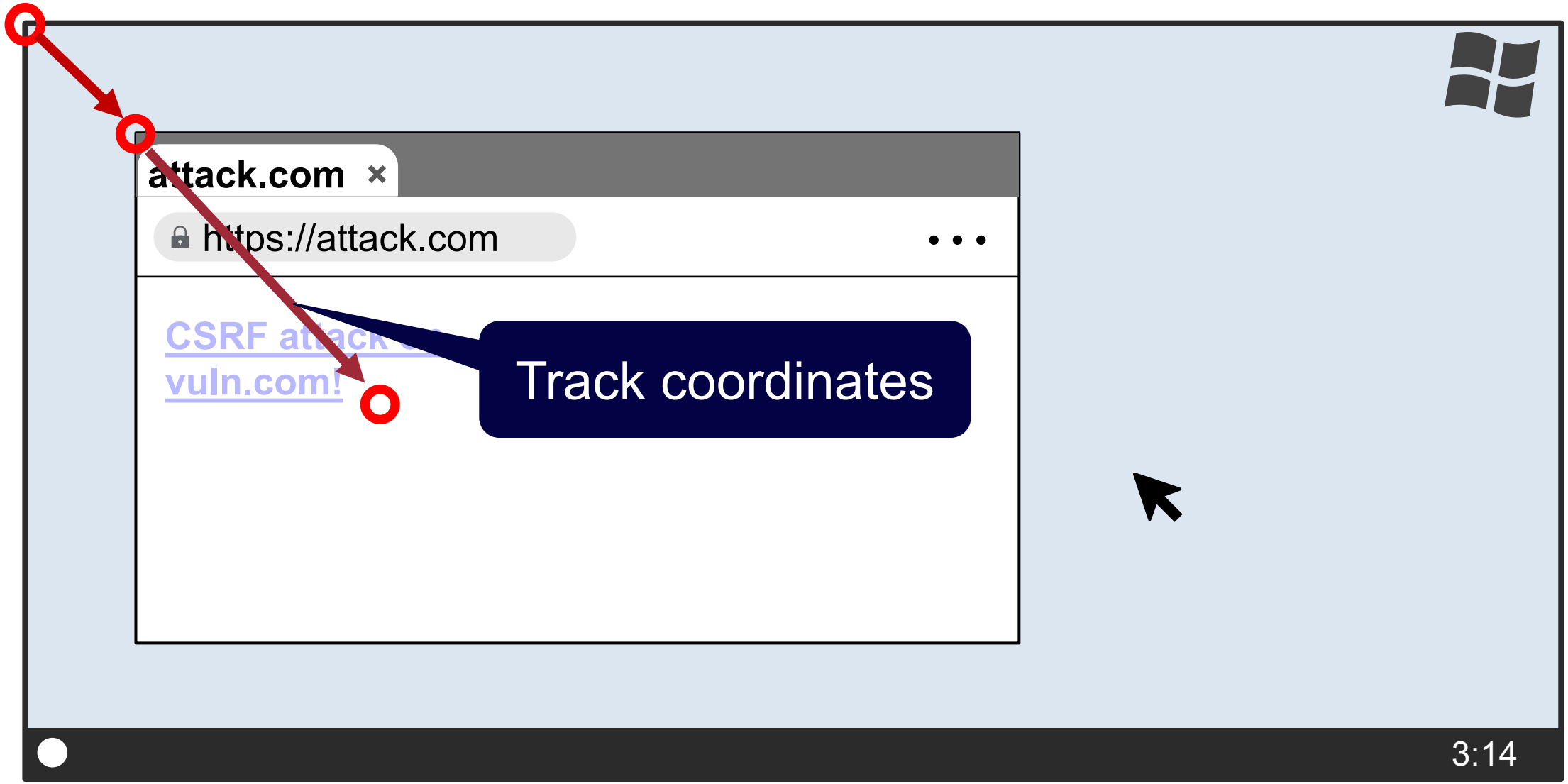
User interaction simulation

Bug oracles implementation

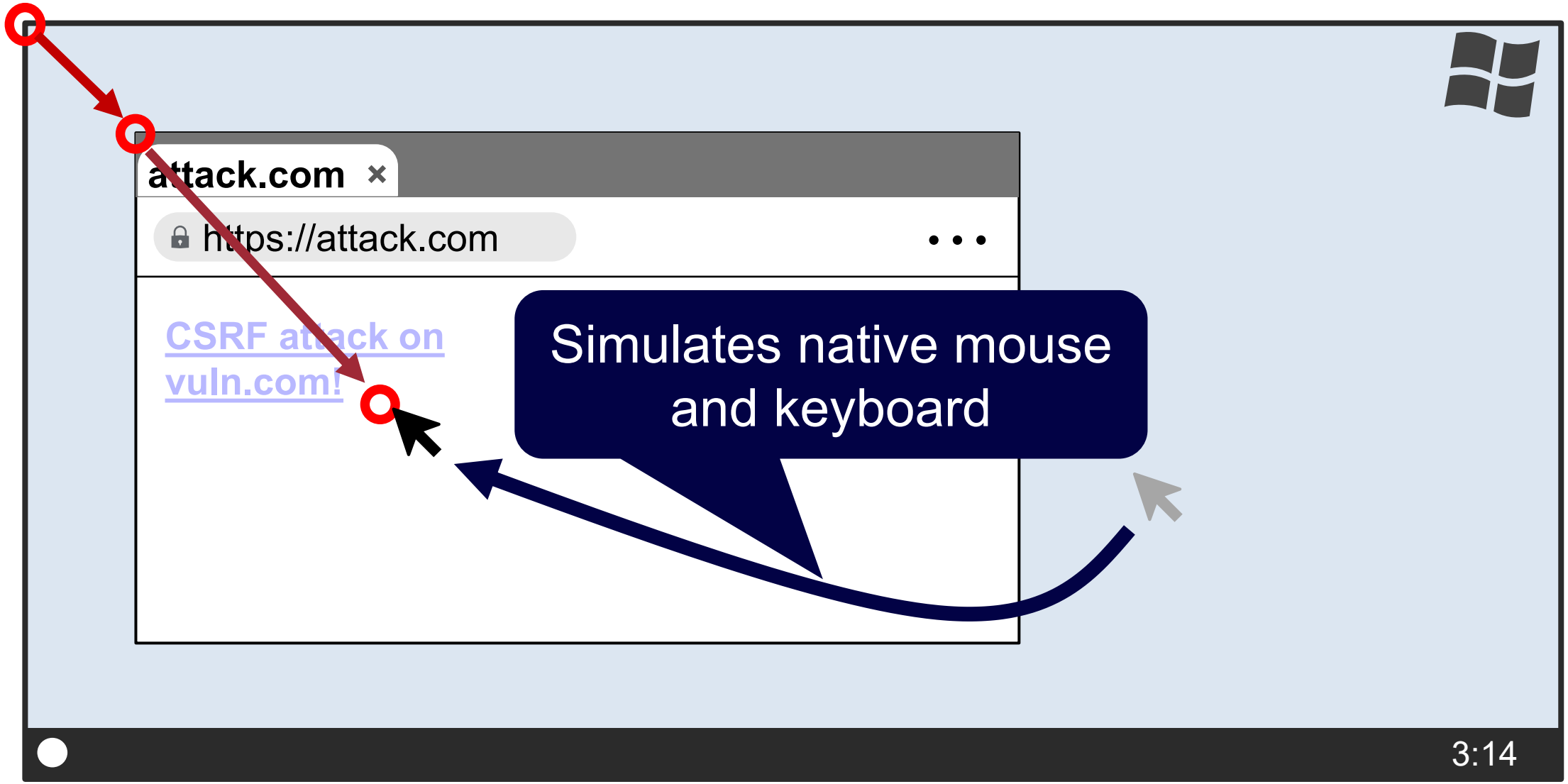
Filtering for navigation-related interactions

Simulating at the OS level

Simulating at the OS Level



Simulating at the OS Level

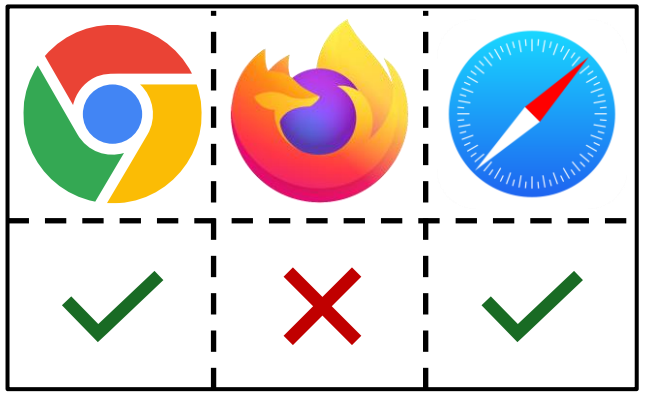


Pre/Post-Interaction Oracle

Large search space

User interaction simulation

Bug oracles implementation

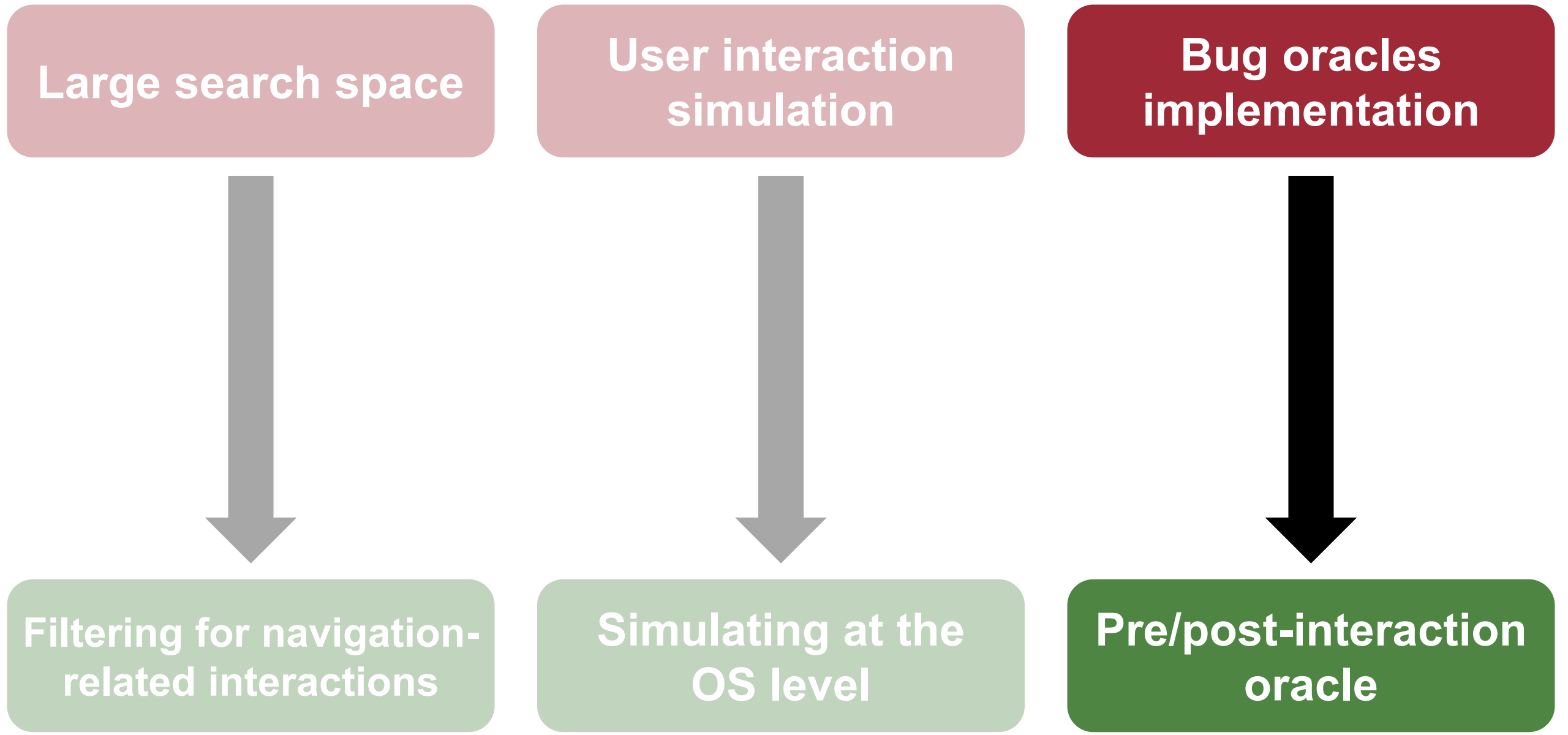


Existing solution:
Cross-browser differential testing

Filtering for navigation-related interactions

Simulating at the OS level

Pre/Post-Interaction Oracle



Pre/Post-Interaction Oracle

Enables testing on a single browser

Initial visit

https://attack.com

CSRF attack on vuln.com!

Click!

vs.

After the interaction

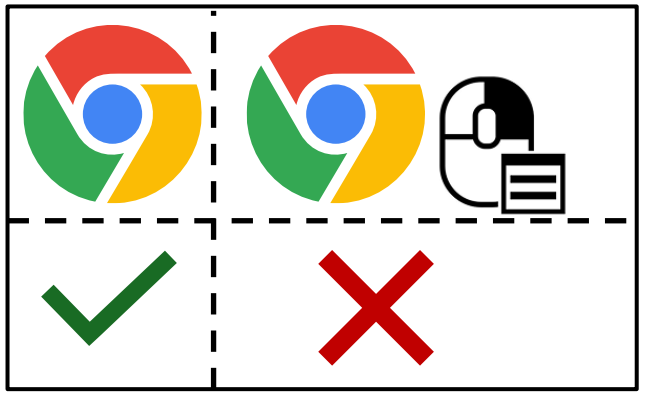
https://attack.com

CSRF attack on vuln

Open link in new tab

Open link in split view

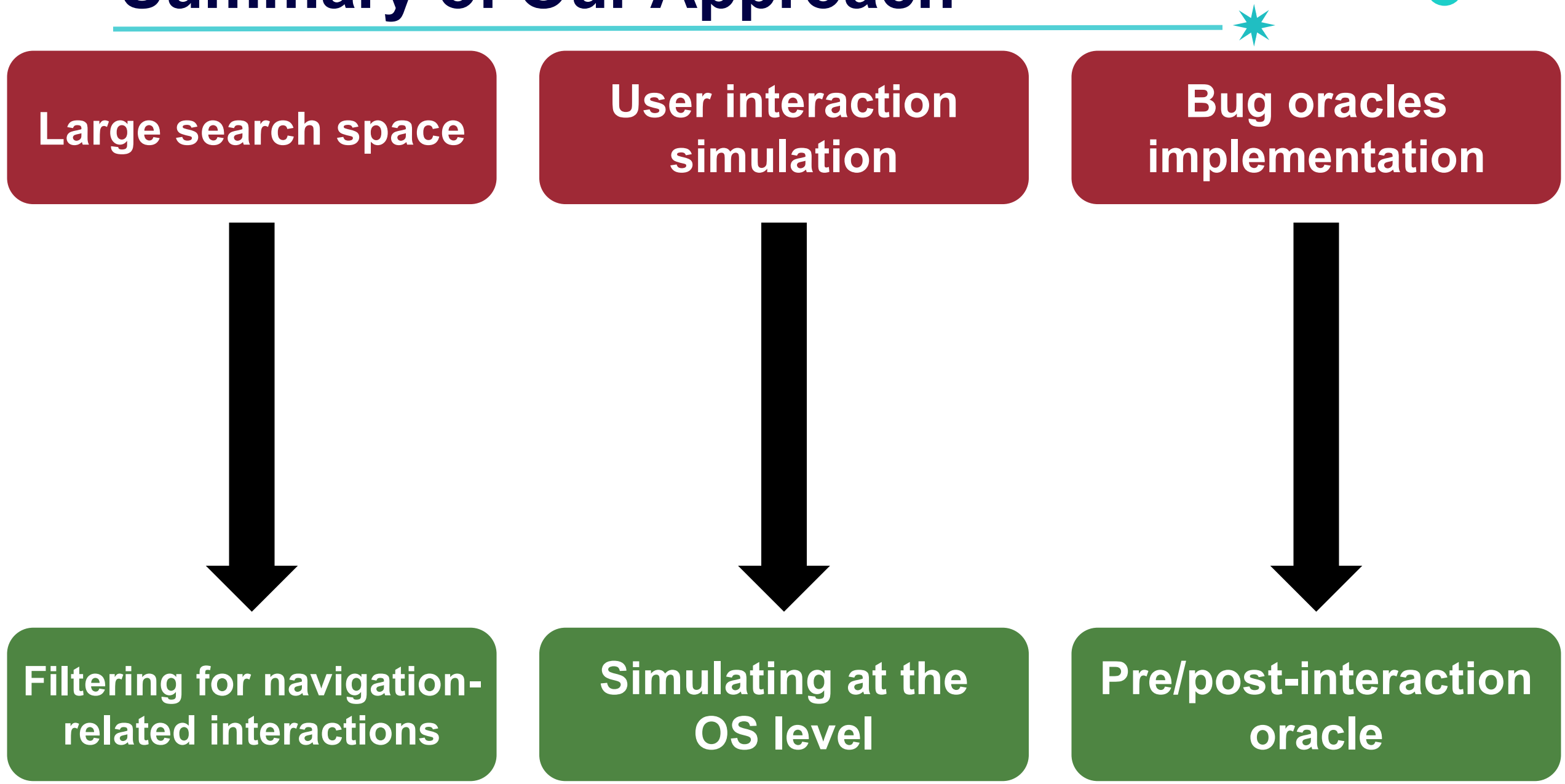
Open link in new window



Our approach

Pre/post-interaction oracle

Summary of Our Approach



Experimental Setup

- **Test 6 browsers :** Chrome, Firefox, Edge, Opera, Brave, Whale
  - ✓ Collected 76 browser UI-level interactions

- **Simulate interactions on the existing test webpages**
  - ✓ Extracted from WPT and prior studies
  - ✓ 13,109 test pages
- **Machines:** 12 desktop machines
- **Total testing time:** 196 hours (28-38 hours per browser)

Evaluation: Bugs Found

- Found **35** security bugs in **6** browsers
  - **14** bugs have been patched and **7** CVEs assigned

Bugs by Browser					
Chrome	Firefox	Edge	Brave	Opera	Whale
8	3	3	3	3	15

Bugs by Security Policy				
CSP	SameSite cookie	PP	COOP	HSTS
25	7	1	1	1

Microsoft NAVER brave

→ **Rewarded with $14,700**

Evaluation: Bugs Found

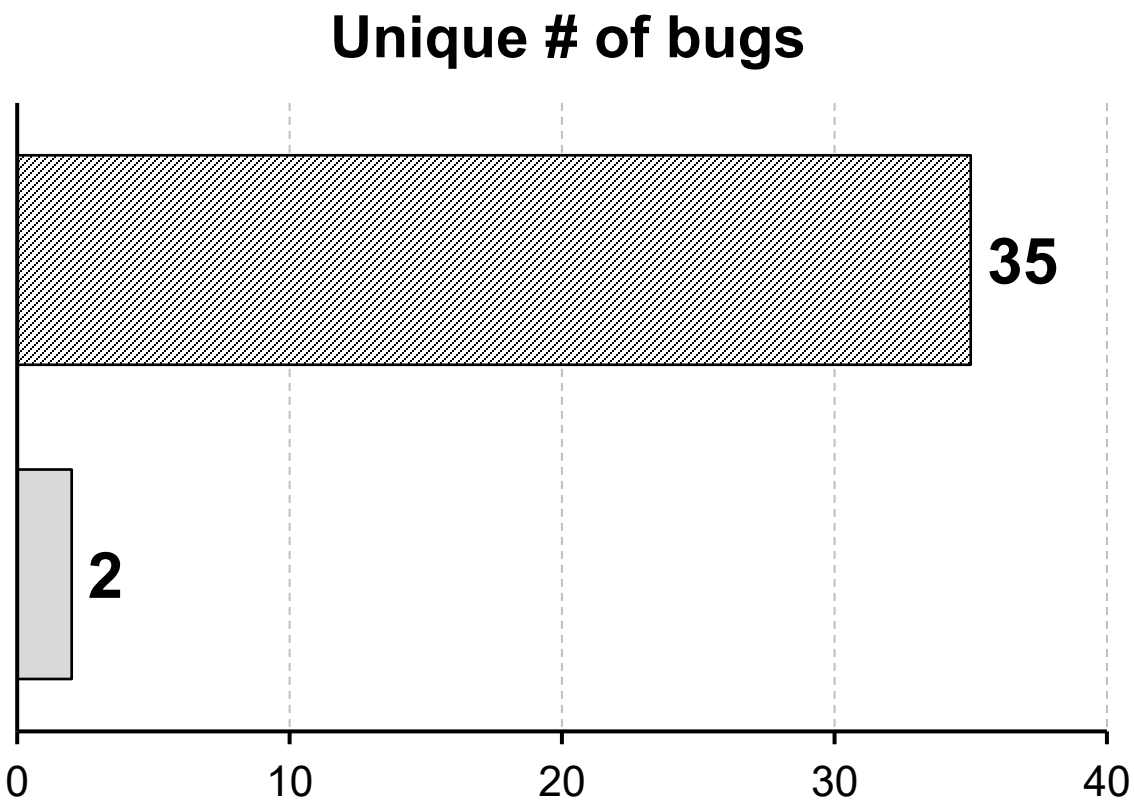
- **2** false positives
  - In SameSite and CSP sandbox enforcement
  - Due to specification-compliant pre/post differences

Evaluation: Bug Oracle

Single-browser testing enables
6× faster bug detection

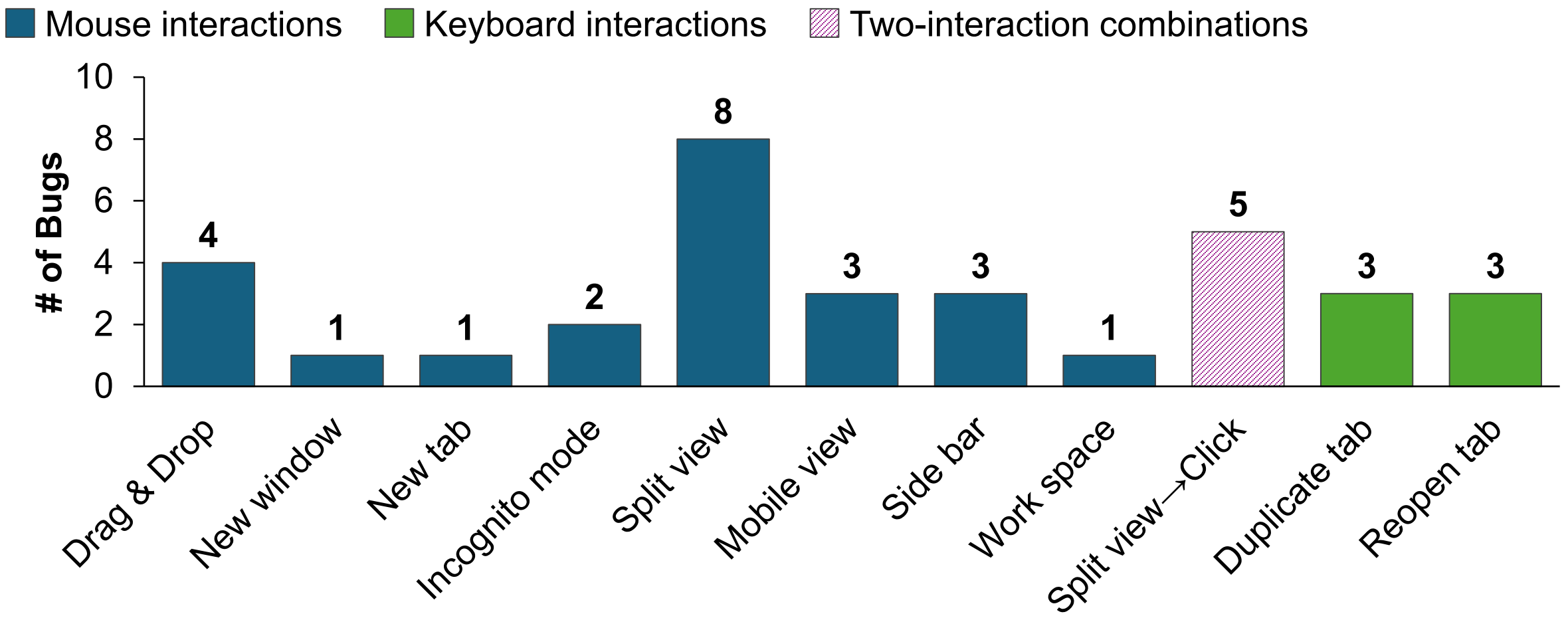
(Ours) Pre/post-interaction oracle

(Prior work) Differential testing oracle

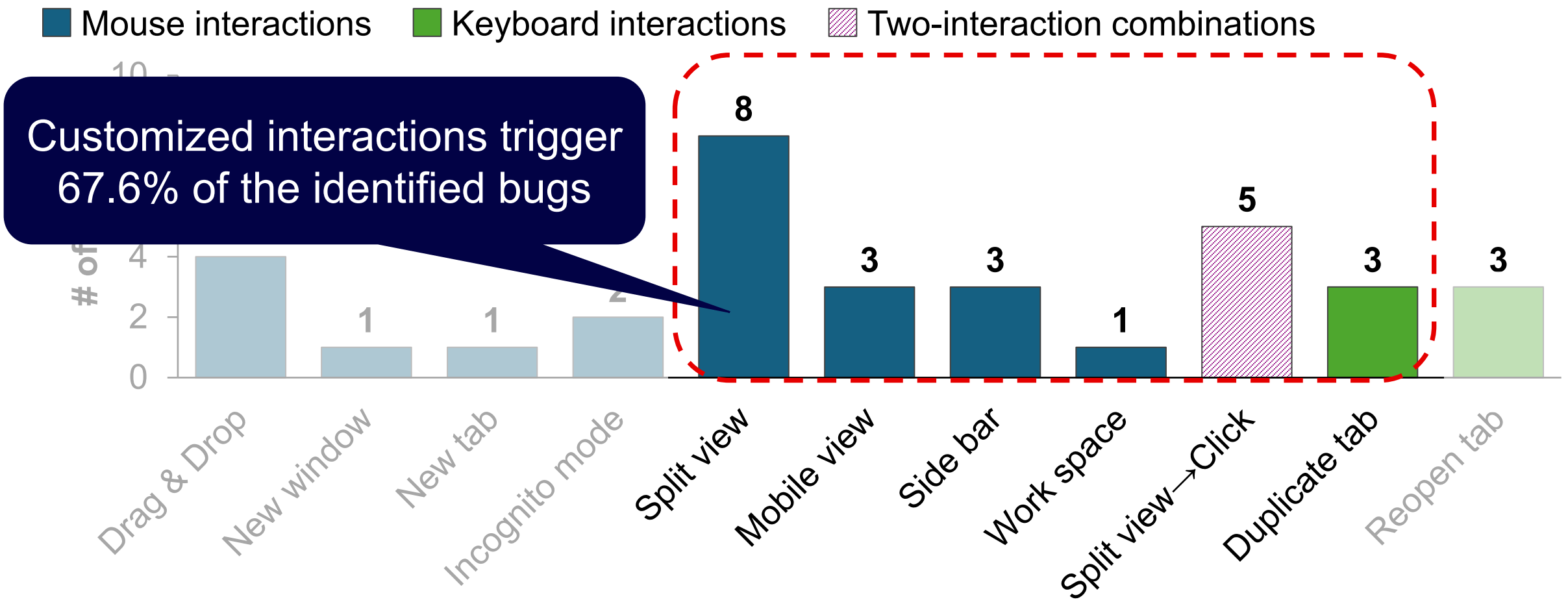


Pre/post-interaction oracle outperforms cross-browser testing

Evaluation: Browser UI-level Interactions



Evaluation: Browser UI-level Interactions



Upstream (e.g., Chromium) security does not cover downstream (e.g., Edge) UI customizations

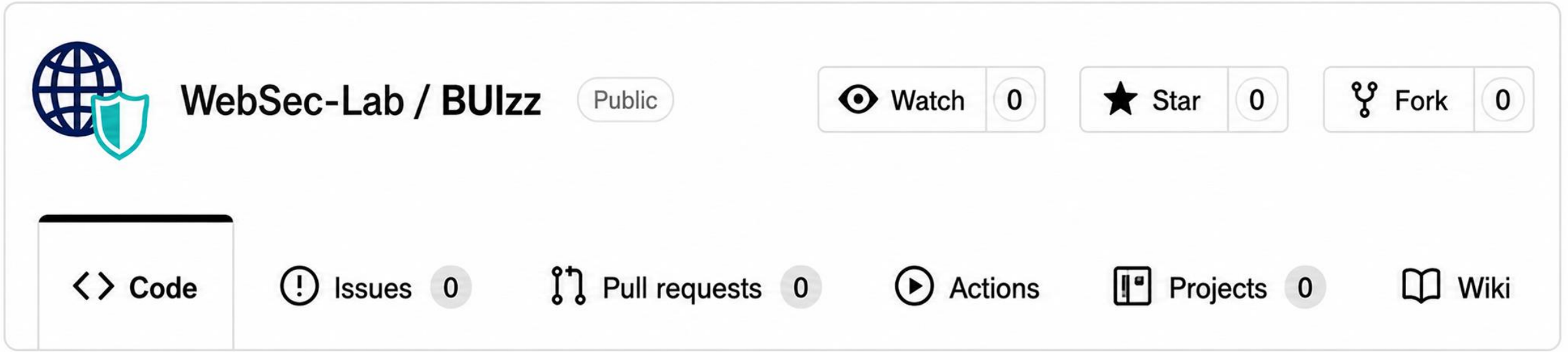
Conclusion

- **New attack surface:** browser UI-level interactions

- **BUIzz:** the first approach to find policy enforcement bugs triggered by browser UI-level interactions
  - Collects interactions **by filtering for navigation-related interactions**
  - Simulates interactions **at the OS level**
  - Detects bugs **by leveraging the pre/post-interaction oracle**

- **Found 35 security bugs**
  - 14 bugs have been patched
  - 7 CVEs assigned

Open Science



https://github.com/WebSec-Lab/BUIzz

Thank You for Listening

If you’re interested, please come by my poster!