CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

German OWASP Day 2024

Double-Edged Crime: How Browser Extensions Fingerprinting Might Endanger Users and Developers Alike

Shubham Agarwal

German OWASP Day 2024 - Leipzig, DE

Browser Extensions - A Primer

Browser Extensions - A Primer

- Third-party client-side add-ons.

Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.

Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.

Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.
  - *Manifest*.

Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.
  - *Manifest*.
  - *Content Scripts*.

Browser Extensions - A Primer

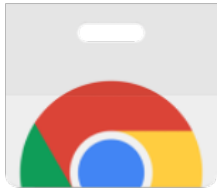
- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.
  - *Manifest.*
  - *Content Scripts.*
  - *Background Scripts/Service Worker.*

Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.
  - *Manifest.*
  - *Content Scripts.*
  - *Background Scripts/Service Worker.*
  - *Extension Popups*

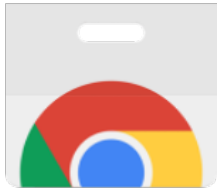
Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.
  - *Manifest.*
  - *Content Scripts.*
  - *Background Scripts/Service Worker.*
  - *Extension Popups*



Browser Extensions - A Primer

- Third-party client-side add-ons.
- Enables various additional features for users.
- Includes different components with different capabilities.
  - *Manifest.*
  - *Content Scripts.*
  - *Background Scripts/Service Worker.*
  - *Extension Popups*



Malwarebytes Browser Guard

www.malwarebytes.com Featured 3.9 ★ (1.1K ratings)

Extension | Privacy & Security | 9,000,000 users

MetaMask

metamask.io 3.0 ★ (4.5K ratings)

Extension | Workflow and planning | 16,000,000 users

Browser Extension Fingerprinting

Browser Extension Fingerprinting

- Cookie-less tracking is on the rise!

Browser Extension Fingerprinting

- Cookie-less tracking is on the rise!
- Major fingerprinting libraries (e.g., FingerprintJS & Castle) have already included extensions in their suite.

Browser Extension Fingerprinting

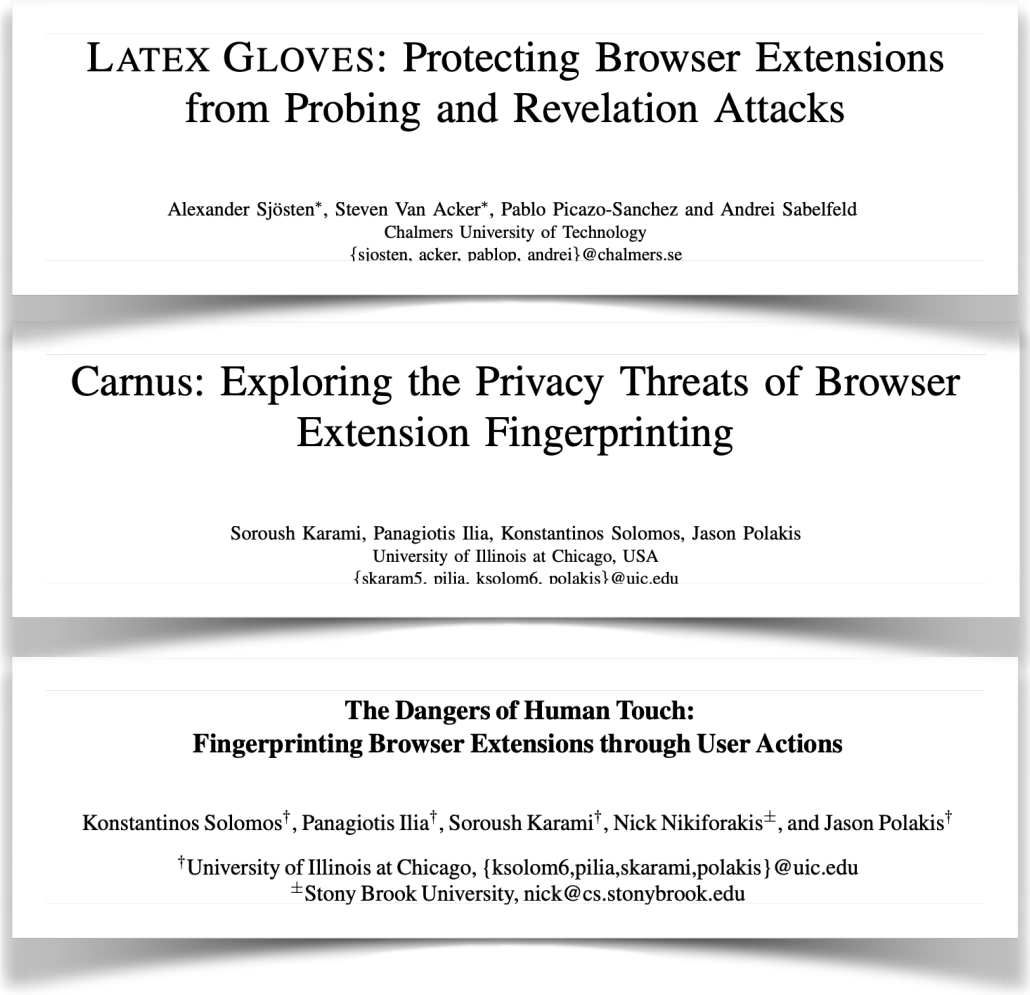
- Cookie-less tracking is on the rise!
- Major fingerprinting libraries (e.g., FingerprintJS & Castle) have already included extensions in their suite.
- Extensions may often exhibit *uniquely identifiable* behavior on the client side.

Browser Extension Fingerprinting

- Cookie-less tracking is on the rise!
- Major fingerprinting libraries (e.g., FingerprintJS & Castle) have already included extensions in their suite.
- Extensions may often exhibit *uniquely identifiable* behavior on the client side.
  - Could be detected by websites to build unique user profiles for tracking.

Browser Extension Fingerprinting

- Cookie-less tracking is on the rise!
- Major fingerprinting libraries (e.g., FingerprintJS & Castle) have already included extensions in their suite.
- Extensions may often exhibit *uniquely identifiable* behavior on the client side.
  - Could be detected by websites to build unique user profiles for tracking.



LATEX GLOVES: Protecting Browser Extensions from Probing and Revelation Attacks

Alexander Sjösten*, Steven Van Acker*, Pablo Picazo-Sanchez and Andrei Sabelfeld
Chalmers University of Technology
{siosten, acker, pablop, andrei}@chalmers.se

Carnus: Exploring the Privacy Threats of Browser Extension Fingerprinting

Soroush Karami, Panagiotis Ilia, Konstantinos Solomos, Jason Polakis
University of Illinois at Chicago, USA
{skaram5, pilia, ksolom6, polakis}@uic.edu

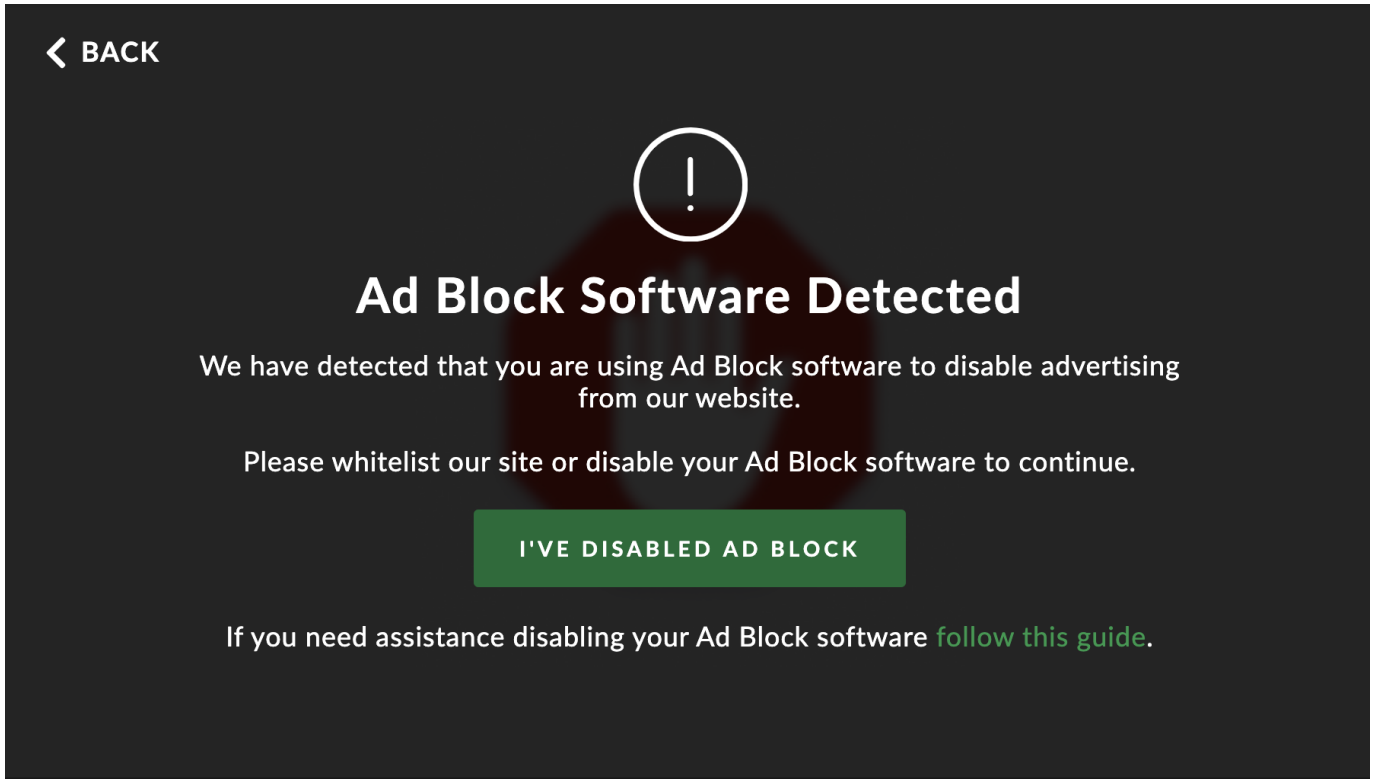
The Dangers of Human Touch:
Fingerprinting Browser Extensions through User Actions

Konstantinos Solomos†, Panagiotis Ilia†, Soroush Karami†, Nick Nikiforakis±, and Jason Polakis†

†University of Illinois at Chicago, {ksolom6,pilia,skarami,polakis}@uic.edu
±Stony Brook University, nick@cs.stonybrook.edu

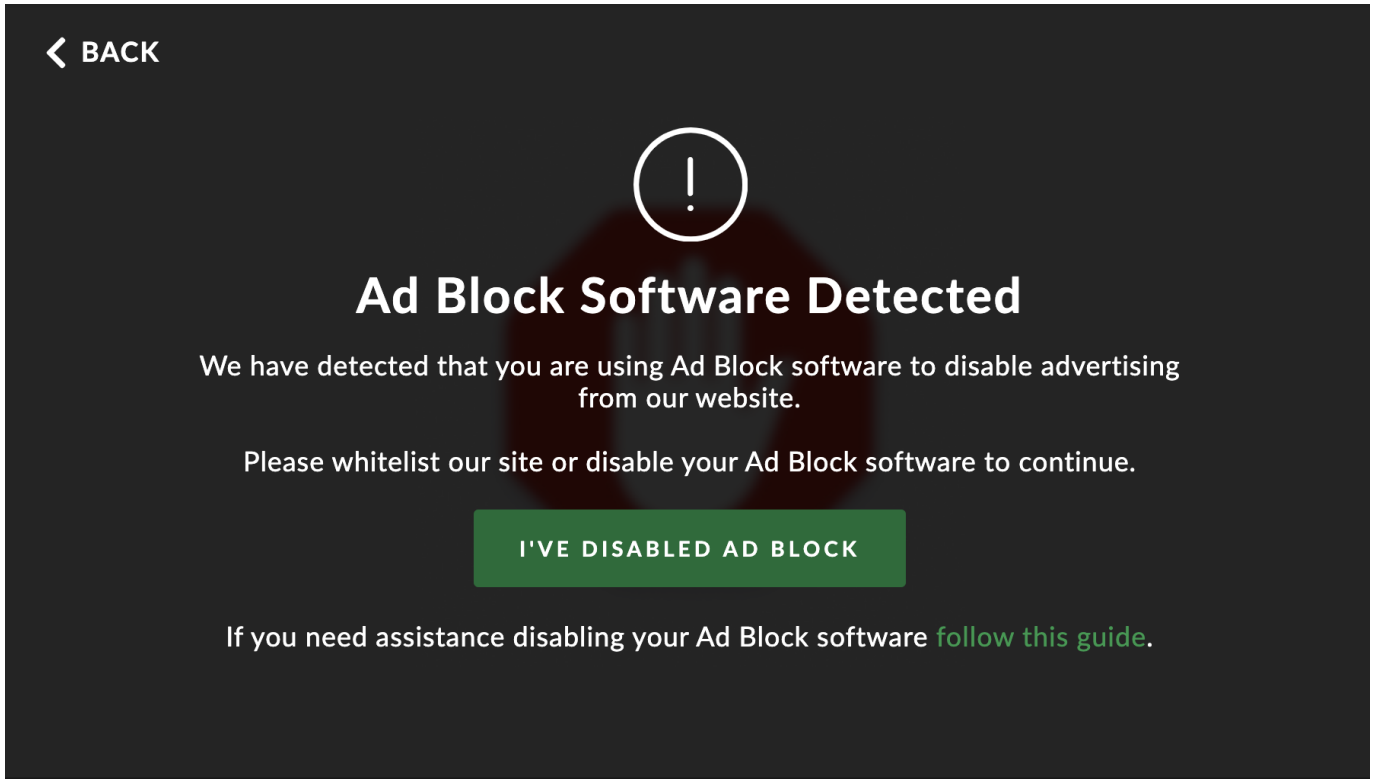
Browser Extension Fingerprinting

Browser Extension Fingerprinting

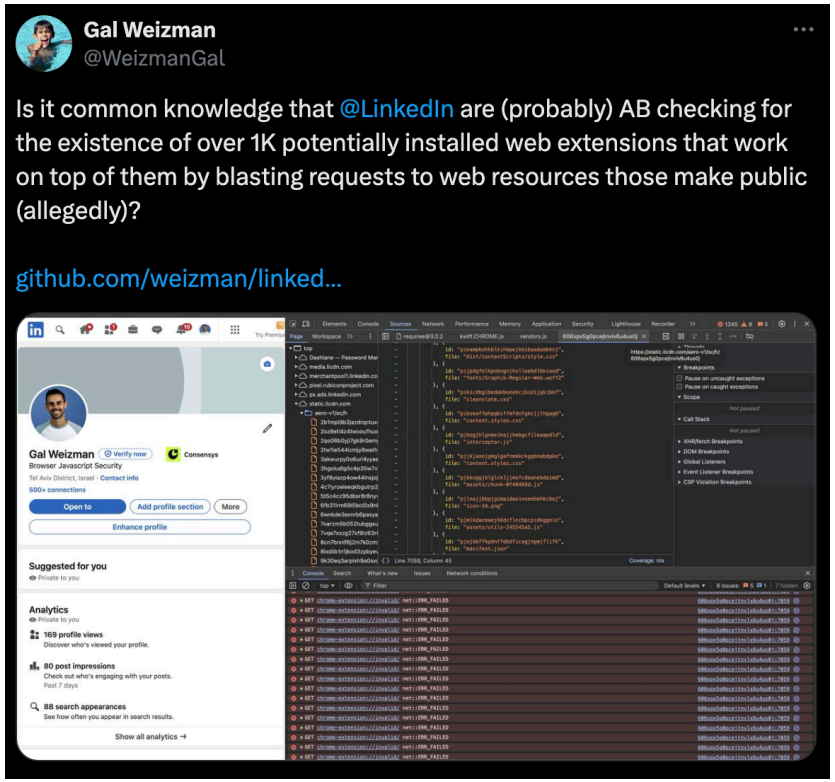


https://www.alphaott.com/iptv-adblock-detection/

Browser Extension Fingerprinting



https://www.alphaott.com/iptv-adblock-detection/



https://x.com/WeizmanGal/status/1840356786114593269

Our Focus – Case I(a)

Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.

Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.

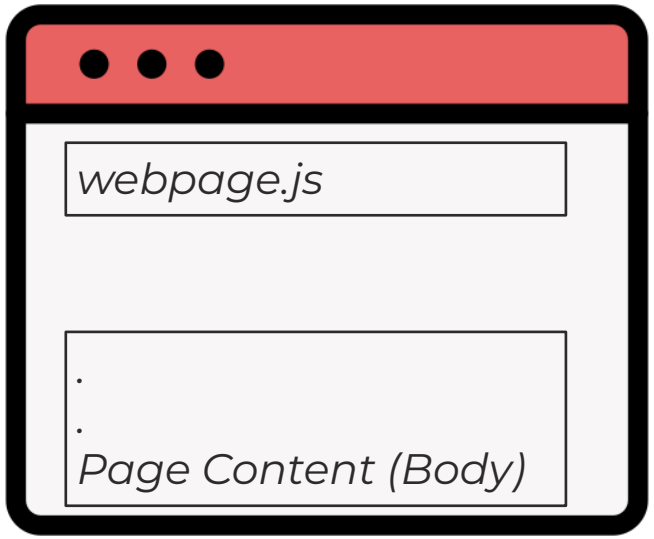
Extension

Website

Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.

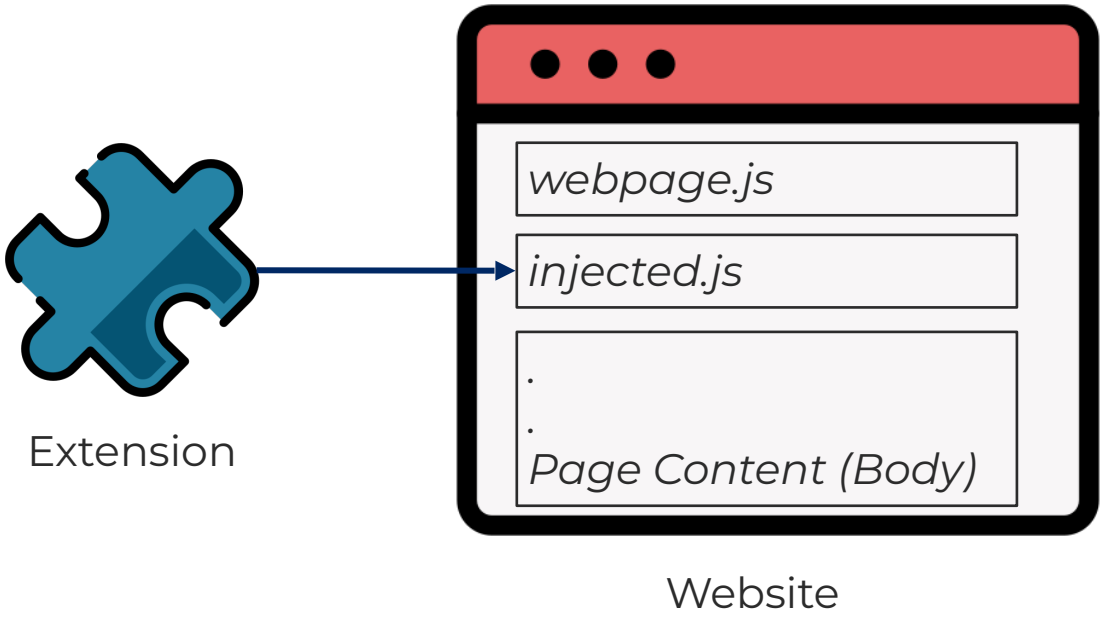
Extension



Website

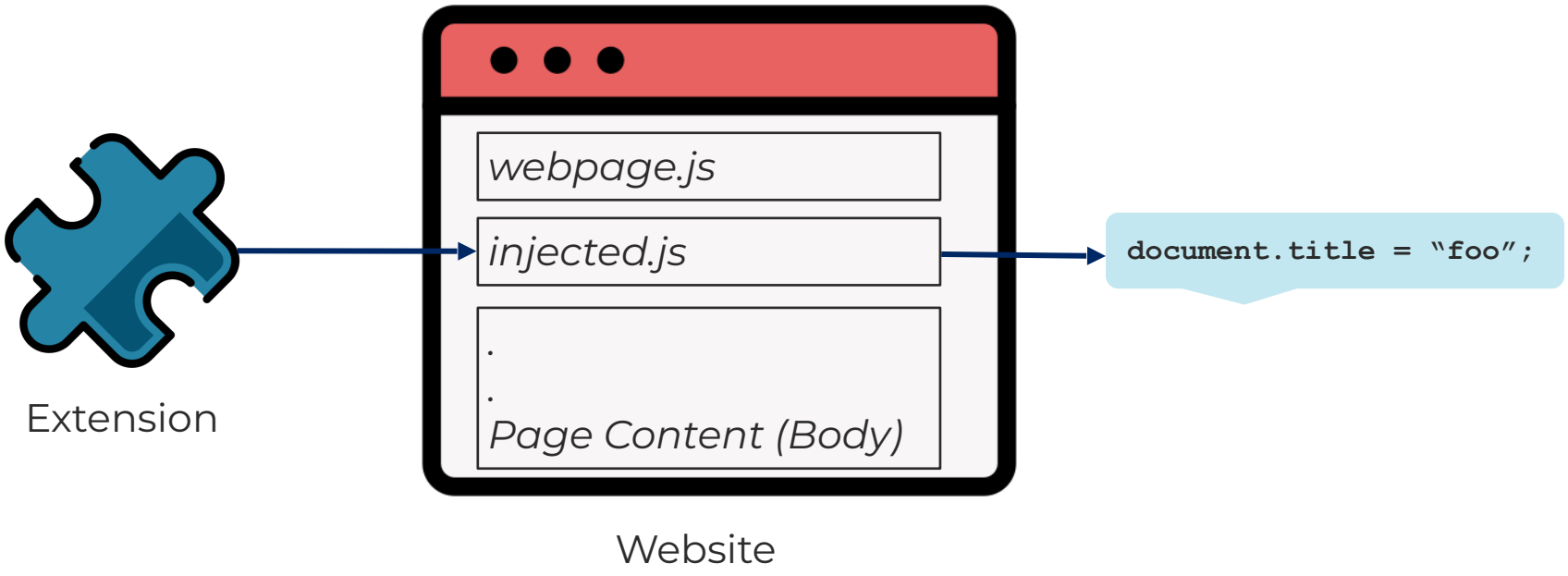
Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.



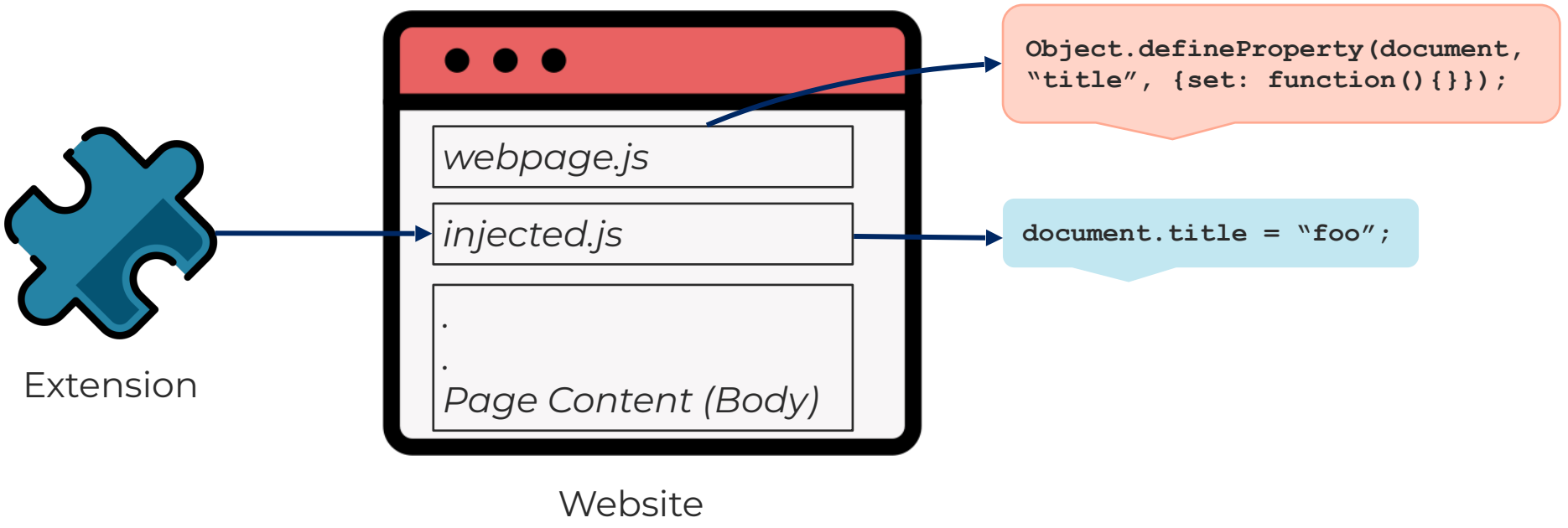
Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.



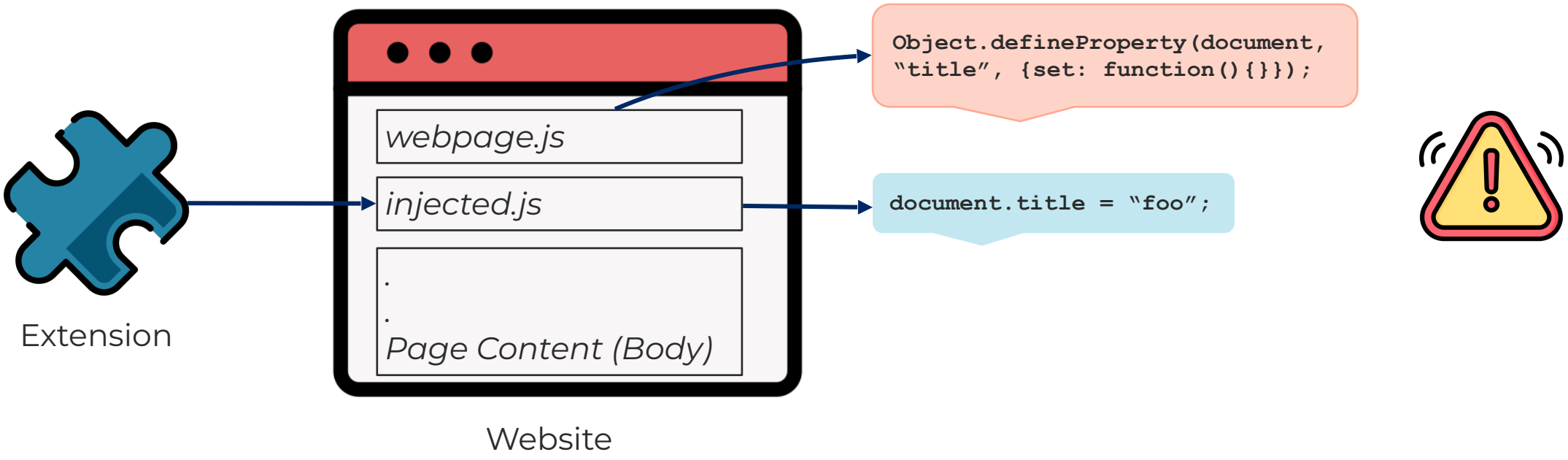
Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.



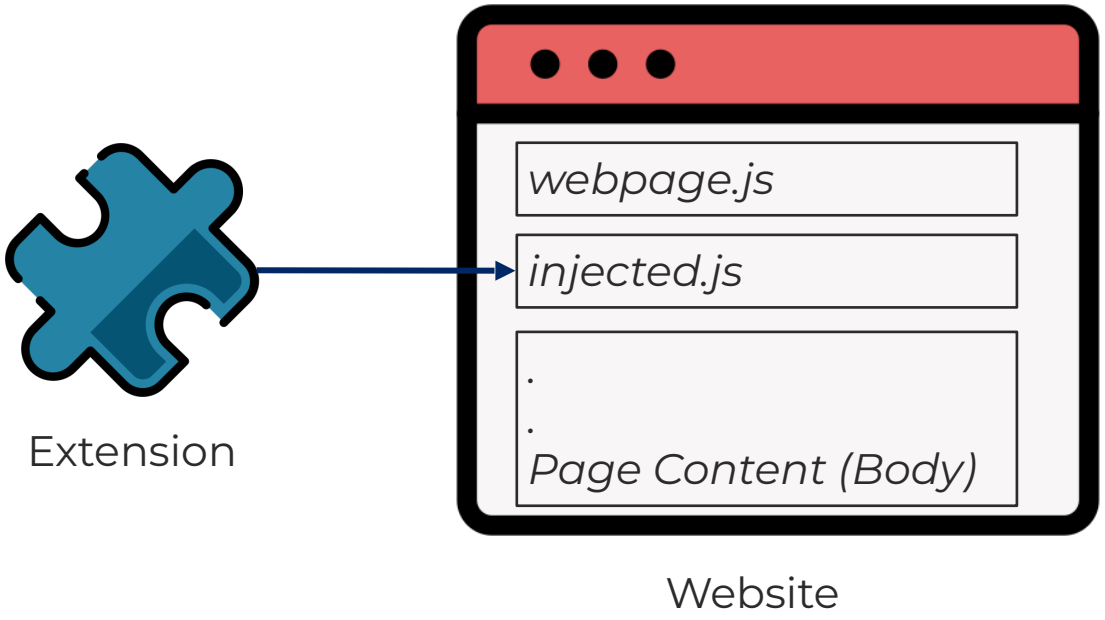
Our Focus – Case I(a)

Extensions often inject JavaScript directly into the visited page.



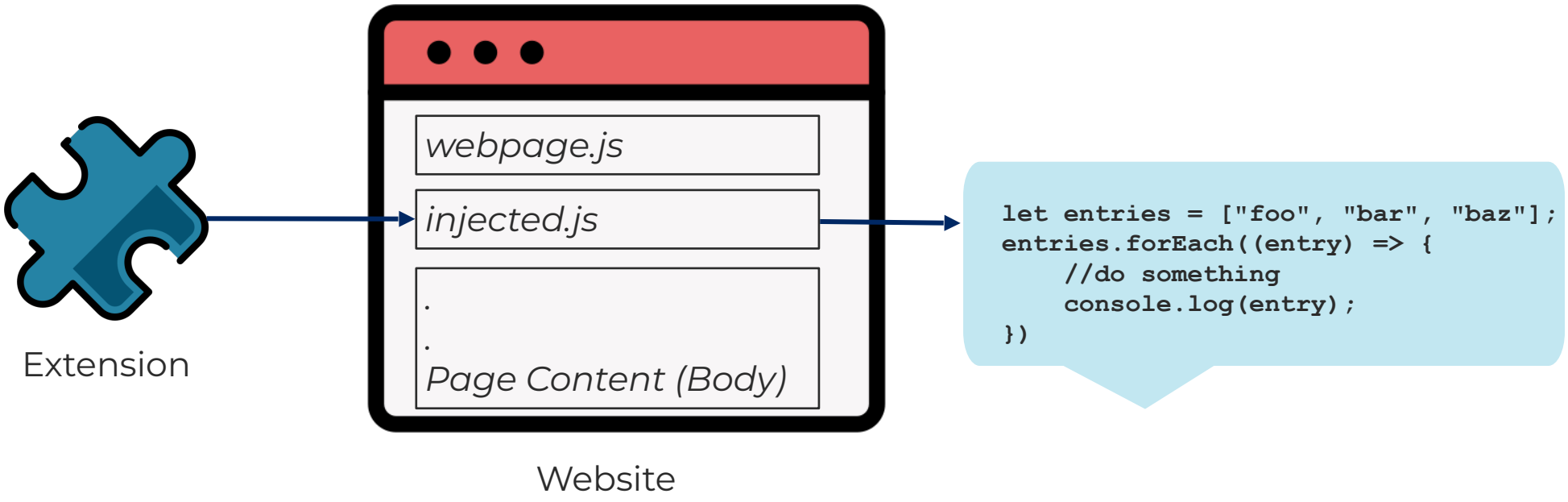
Our Focus – Case I(b)

Extensions often inject JavaScript directly into the visited page.



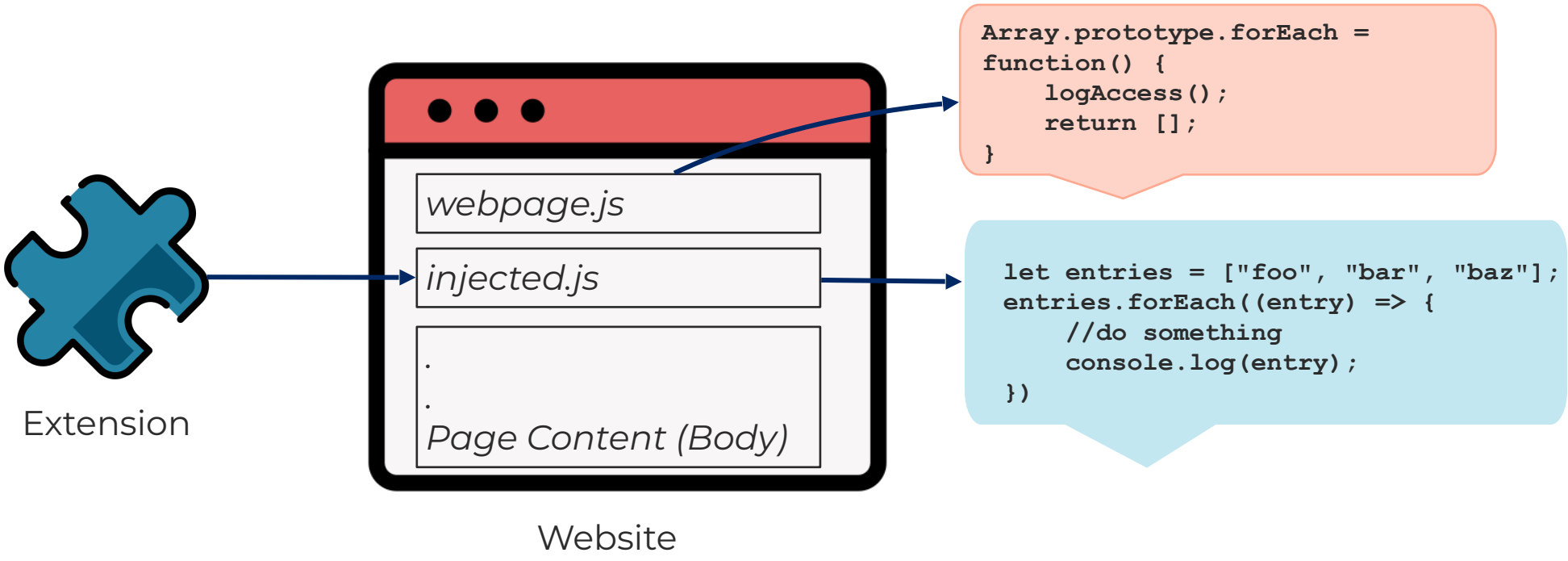
Our Focus – Case I(b)

Extensions often inject JavaScript directly into the visited page.



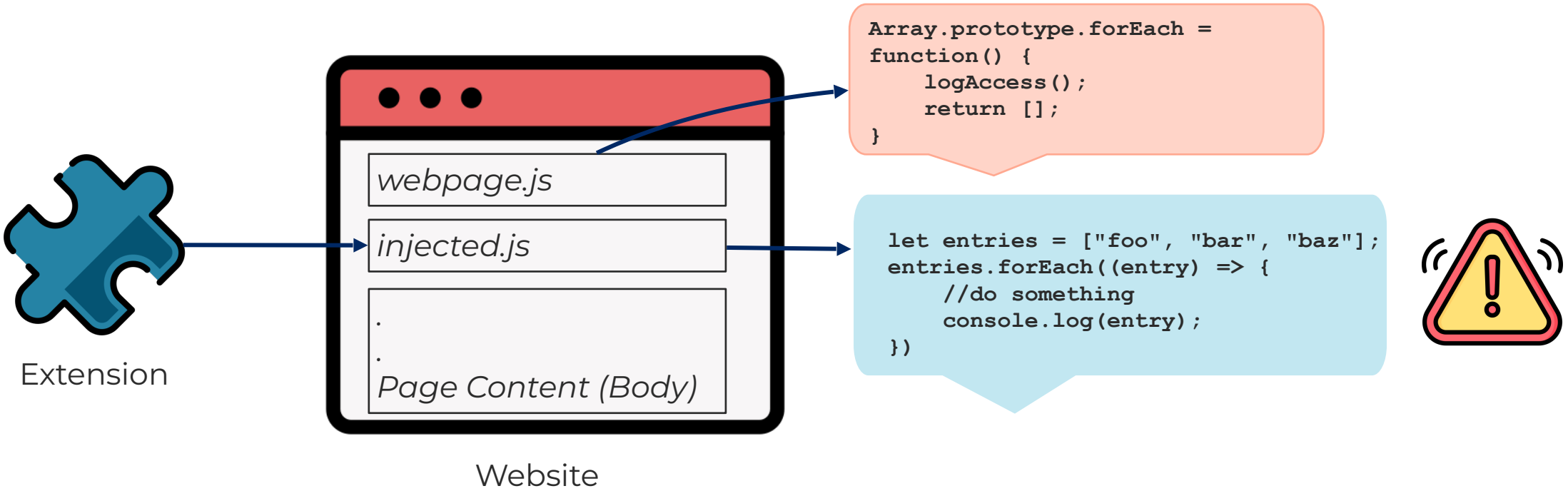
Our Focus – Case I(b)

Extensions often inject JavaScript directly into the visited page.



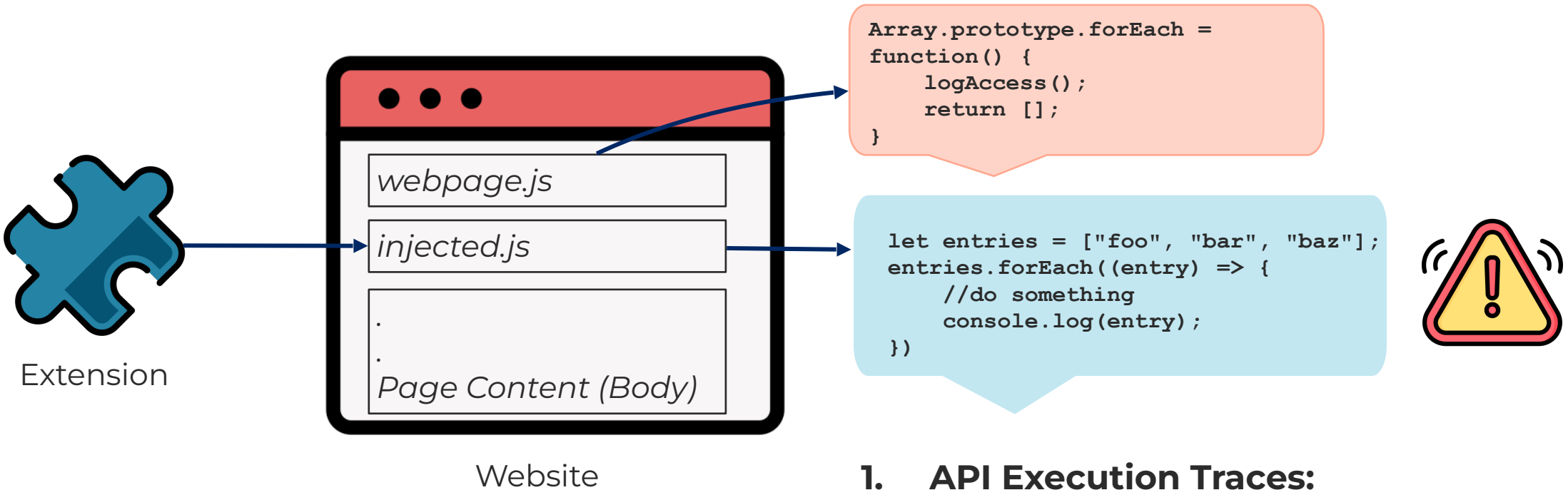
Our Focus – Case I(b)

Extensions often inject JavaScript directly into the visited page.



Our Focus – Case I(b)

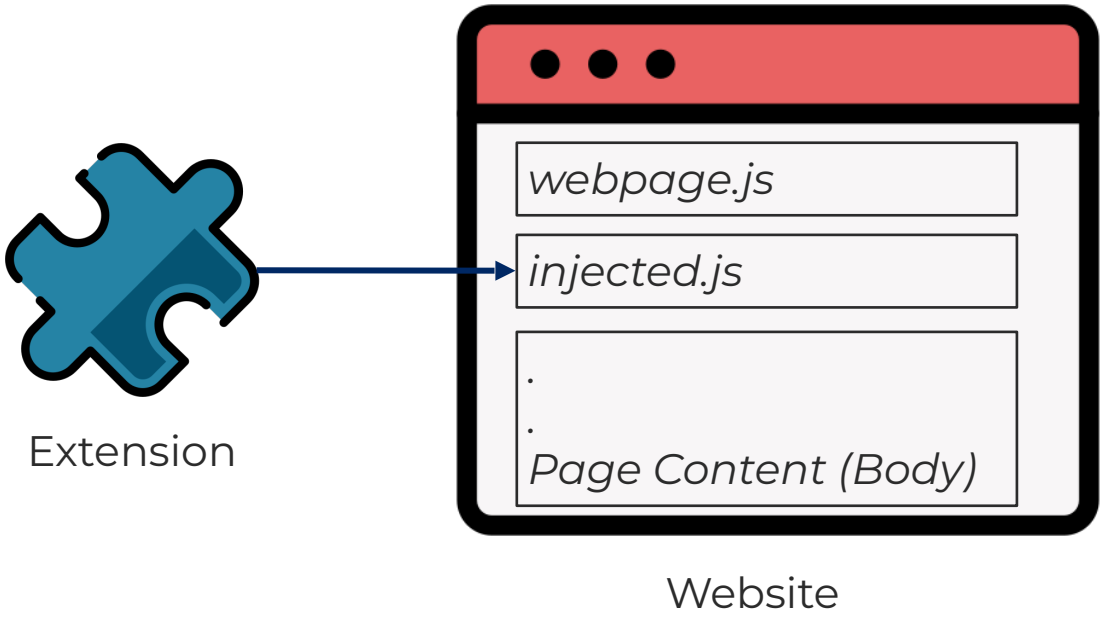
Extensions often inject JavaScript directly into the visited page.



1. **API Execution Traces:**
   i. API Caller Code & Parameters.
   ii. Stack trace of executed API.

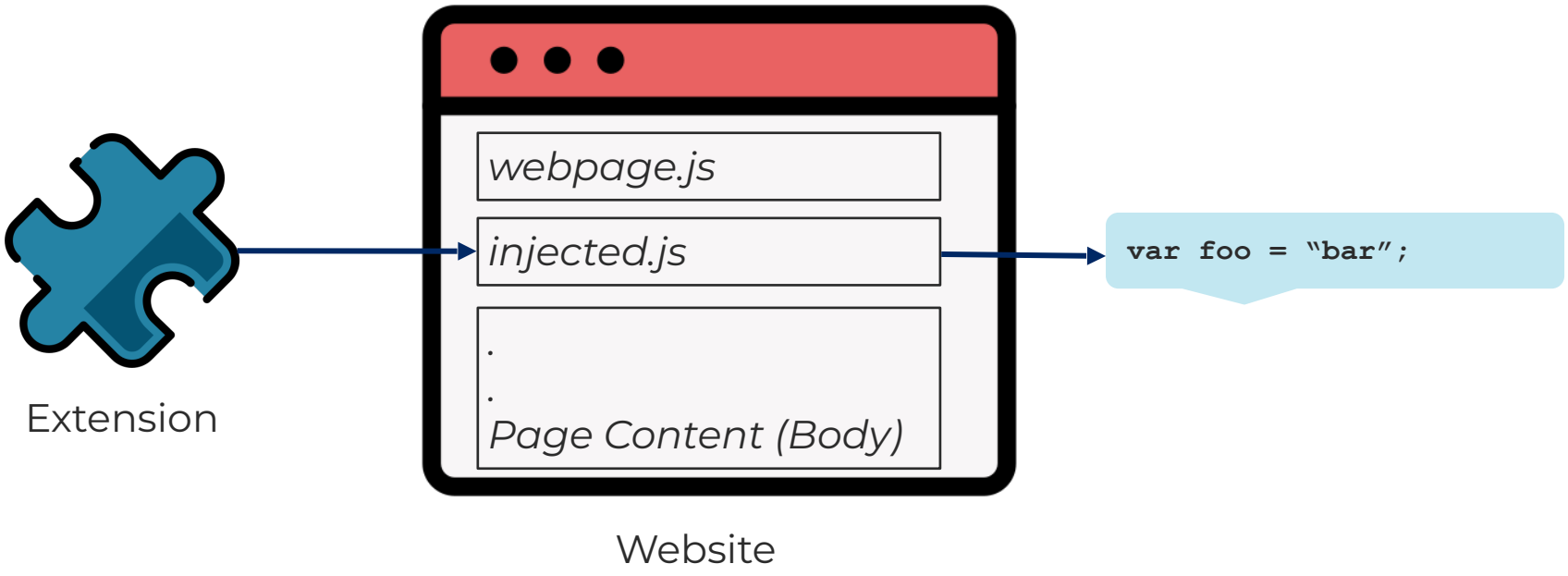
Our Focus – Case I(c)

Extensions often inject JavaScript directly into the visited page.



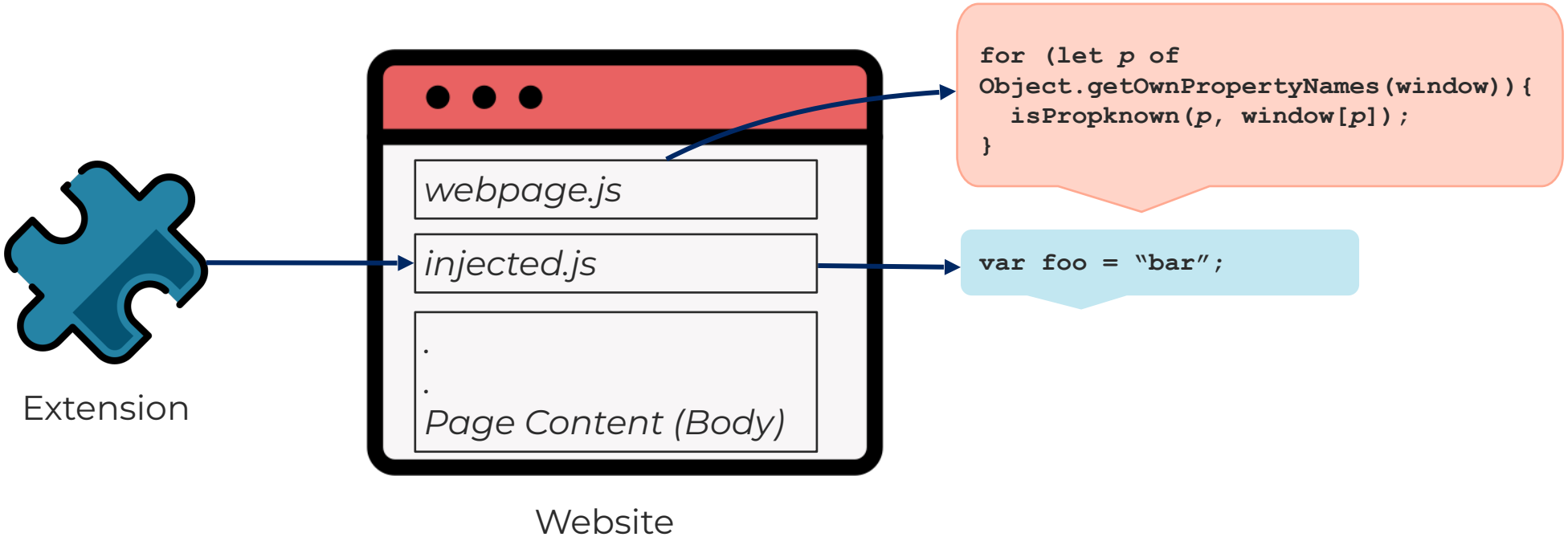
Our Focus – Case I(c)

Extensions often inject JavaScript directly into the visited page.



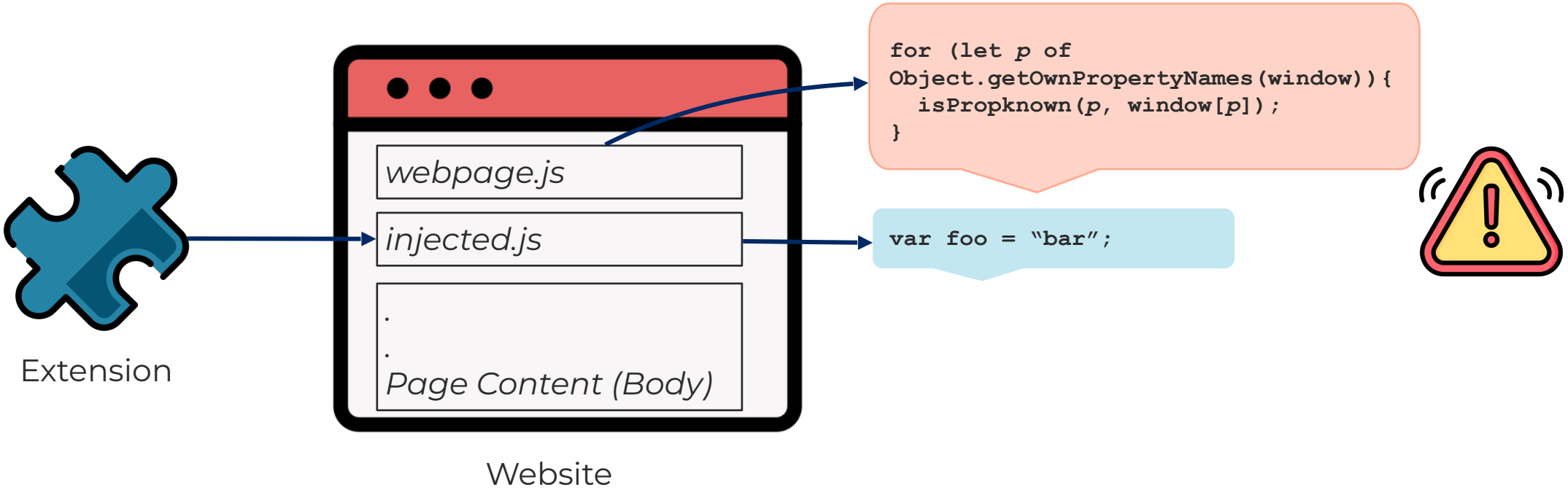
Our Focus – Case I(c)

Extensions often inject JavaScript directly into the visited page.



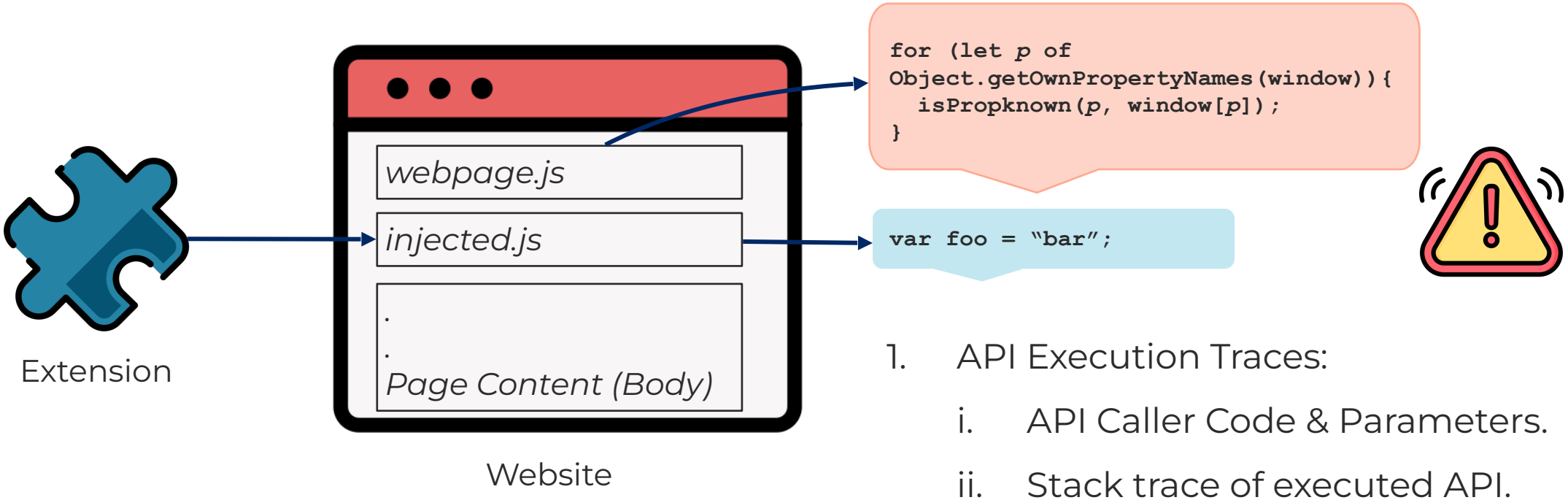
Our Focus – Case I(c)

Extensions often inject JavaScript directly into the visited page.



Our Focus – Case I(c)

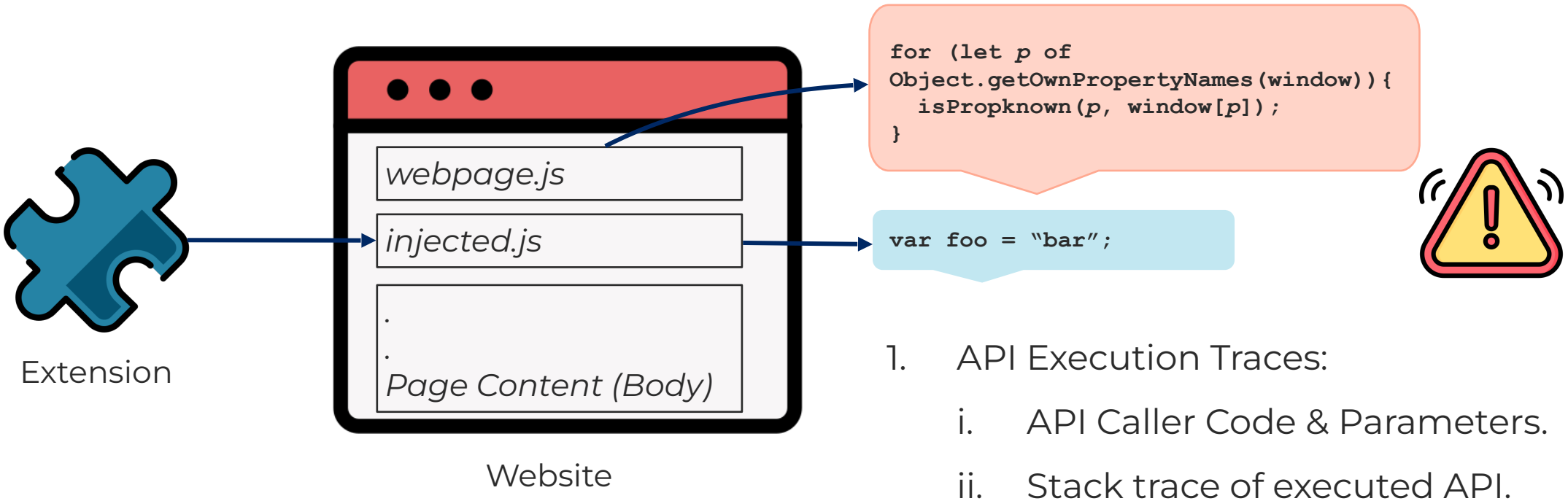
Extensions often inject JavaScript directly into the visited page.



1. API Execution Traces:
   i. API Caller Code & Parameters.
   ii. Stack trace of executed API.
2. **Registered Global Variables.**

Our Focus – Case I(c)

Extensions often inject JavaScript directly into the visited page.



1. API Execution Traces:
   i. API Caller Code & Parameters.
   ii. Stack trace of executed API.
2. **Registered Global Variables.**

Websites can observe the execution of extension-injected code in their own namespace.

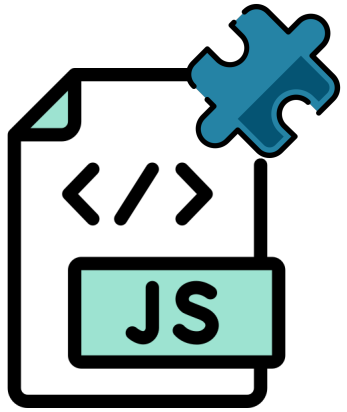
Our Focus – Case II

Our Focus – Case II

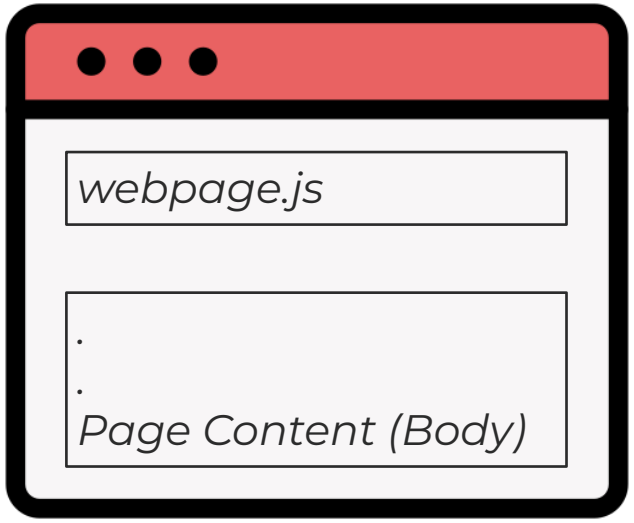
Extensions also store data on the client side through different Storage APIs:

Our Focus – Case II

Extensions also store data on the client side through different Storage APIs:



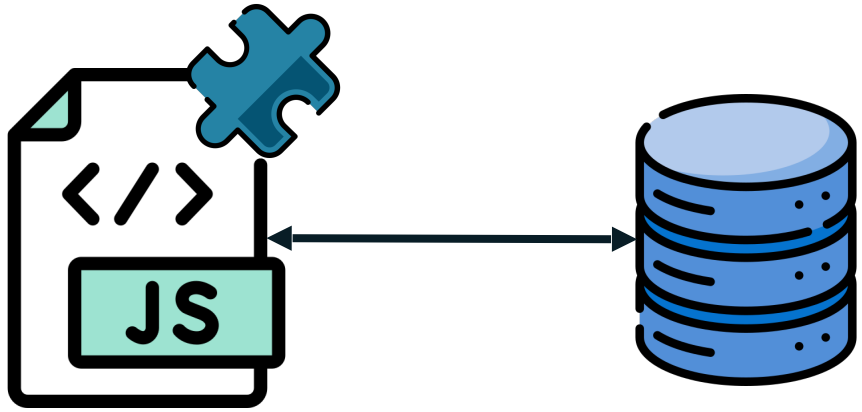
Content Script



Website

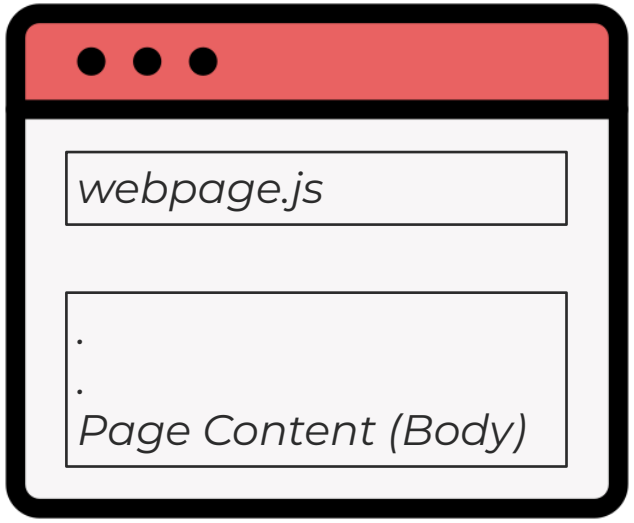
Our Focus – Case II

Extensions also store data on the client side through different Storage APIs:



Content Script

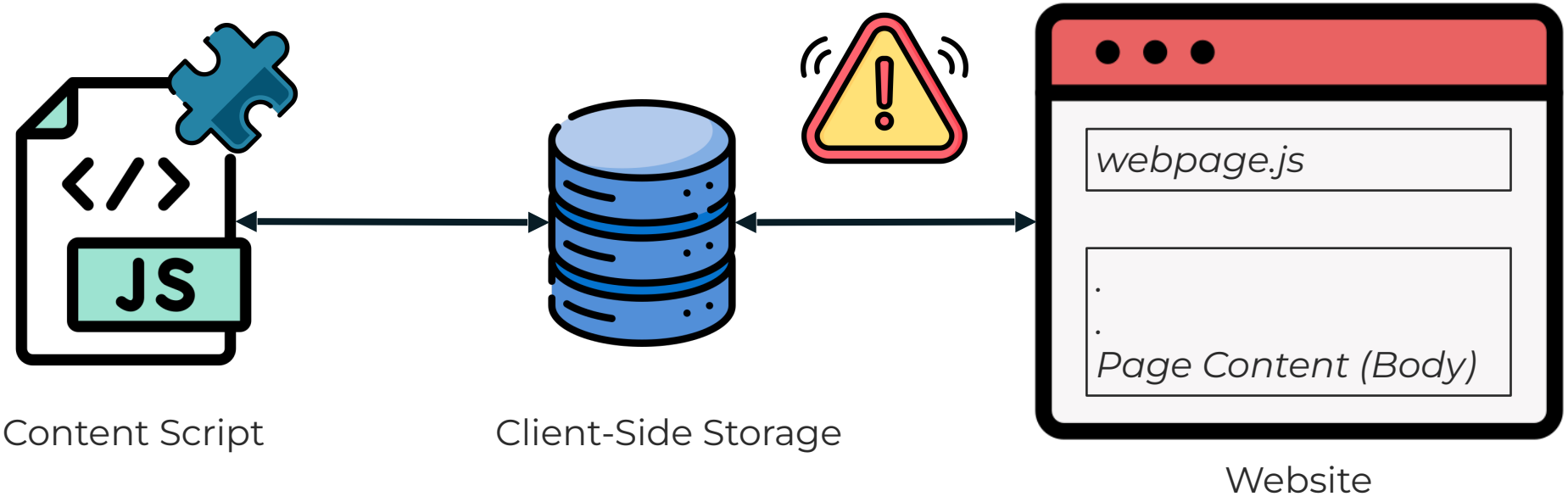
Client-Side Storage



Website

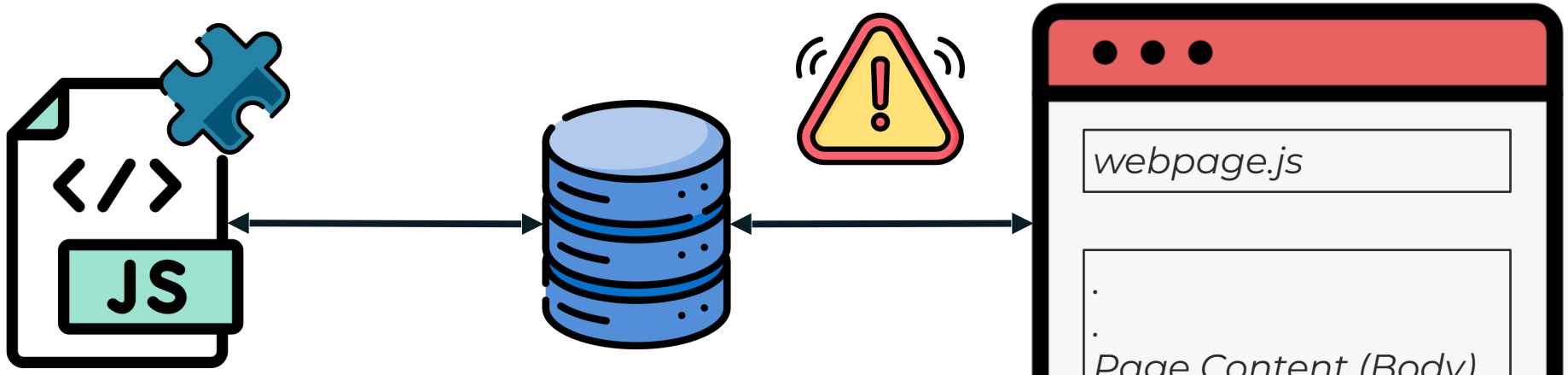
Our Focus – Case II

Extensions also store data on the client side through different Storage APIs:



Our Focus – Case II

Extensions also store data on the client side through different Storage APIs:



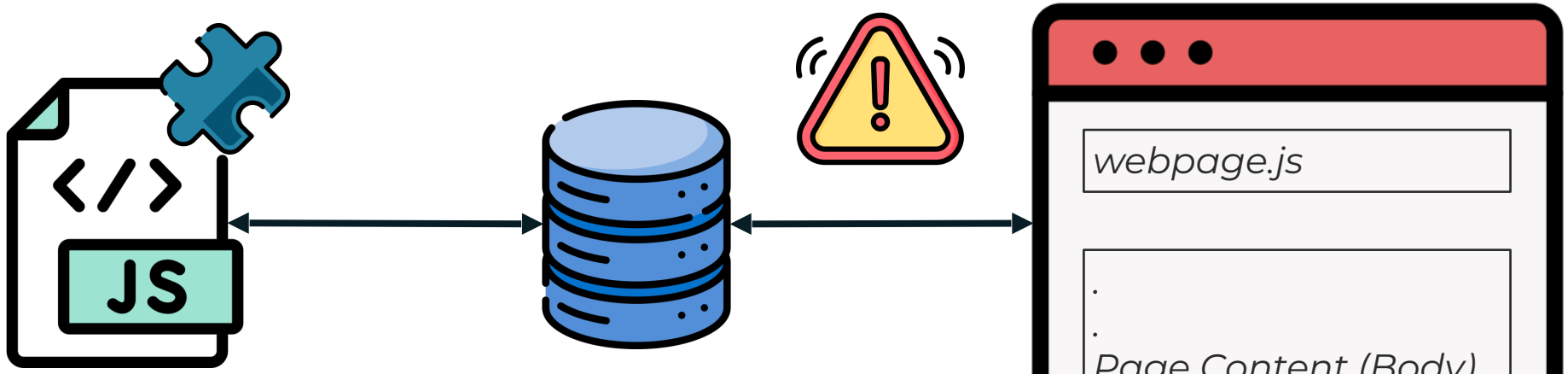
1. Client-side Storage APIs.

   Example: *Local Storage*, *Session Storage*, *IndexedDB* (and *Cookies*).

2. The `postMessage` API.

Our Focus – Case II

Extensions also store data on the client side through different Storage APIs:



1. Client-side Storage APIs.

   Example: *Local Storage*, *Session Storage*, *IndexedDB* (and *Cookies*).

2. The `postMessage` API.

Website can poll through client-side storage APIs to observe extension-induced actions.

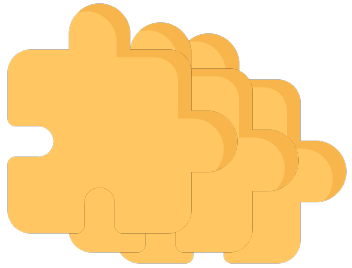
Raider: Pipeline

Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?

Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?



156K Extensions under test

Raider: Pipeline

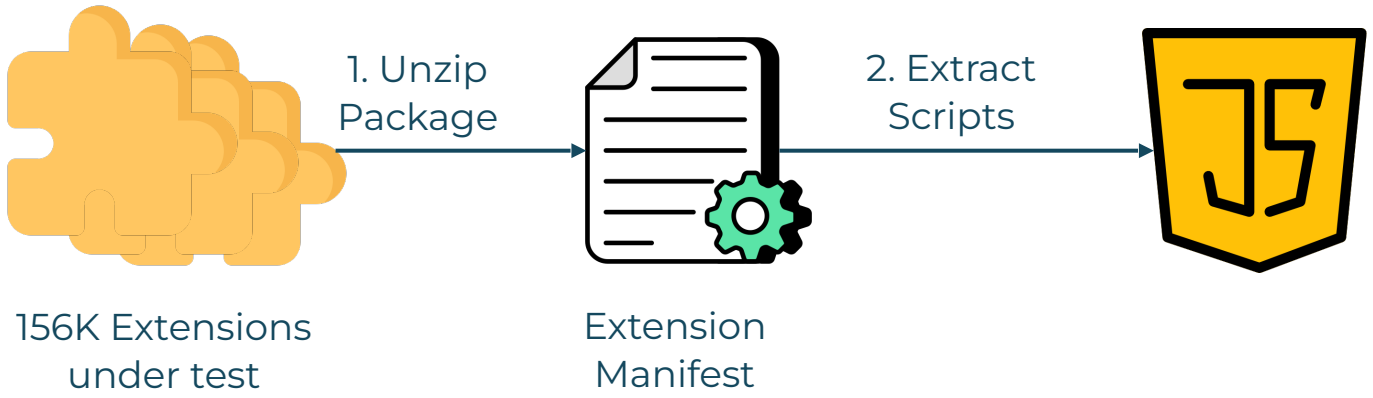
How many extensions can be uniquely fingerprinted through these observable side effects?

156K Extensions under test

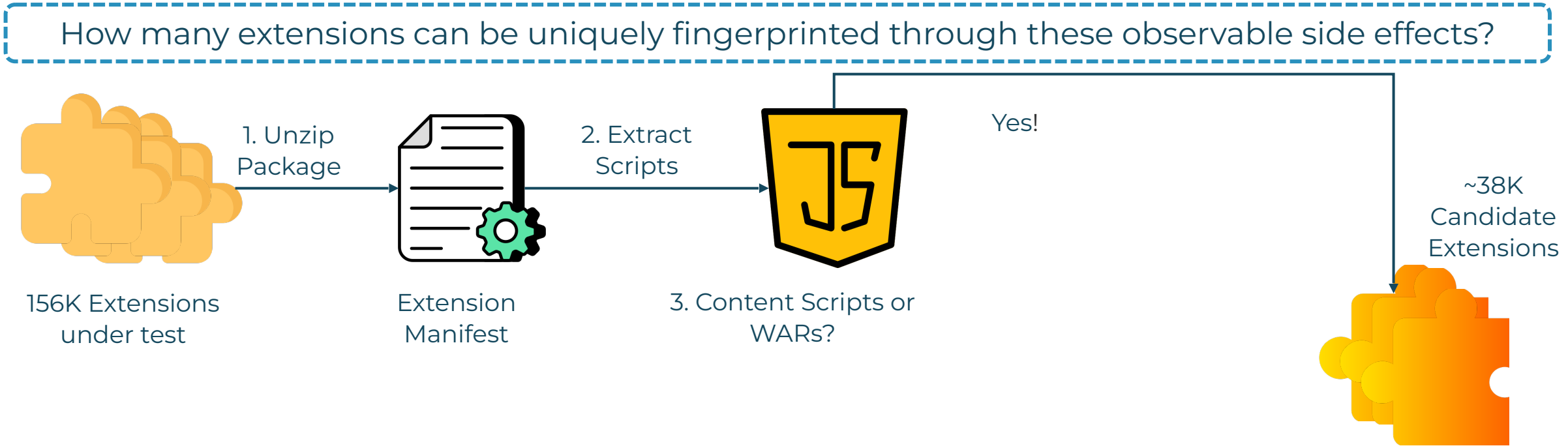
Extension Manifest

Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?

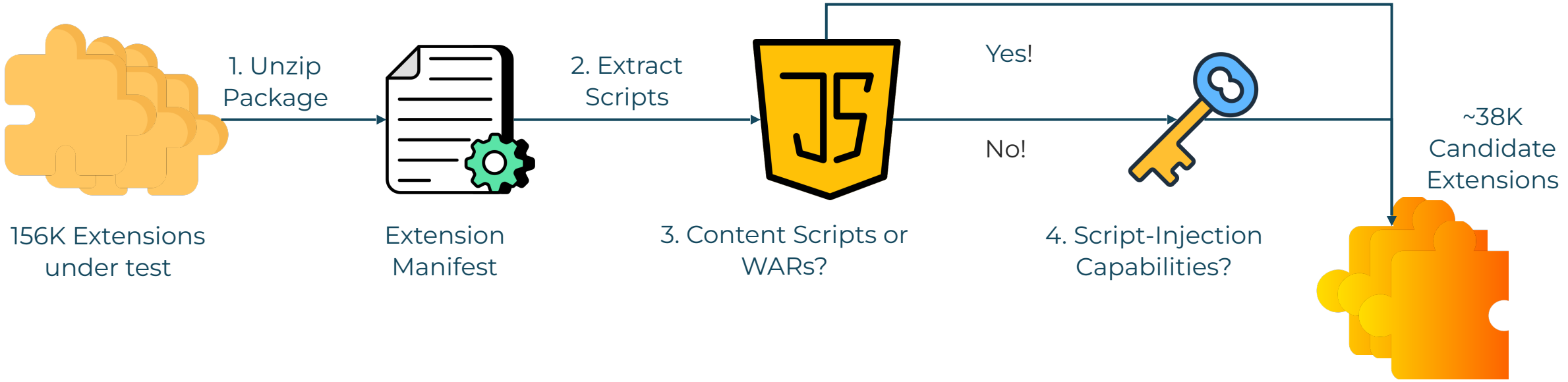


Raider: Pipeline



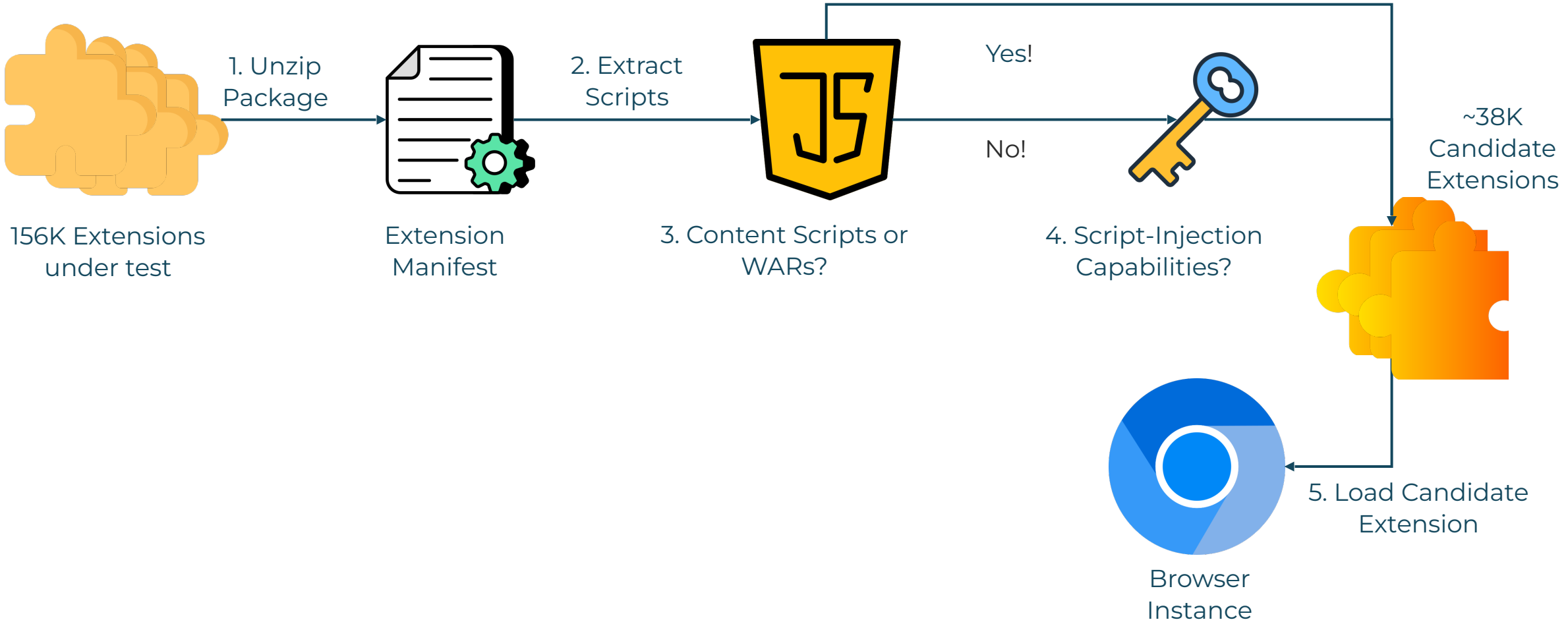
Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?



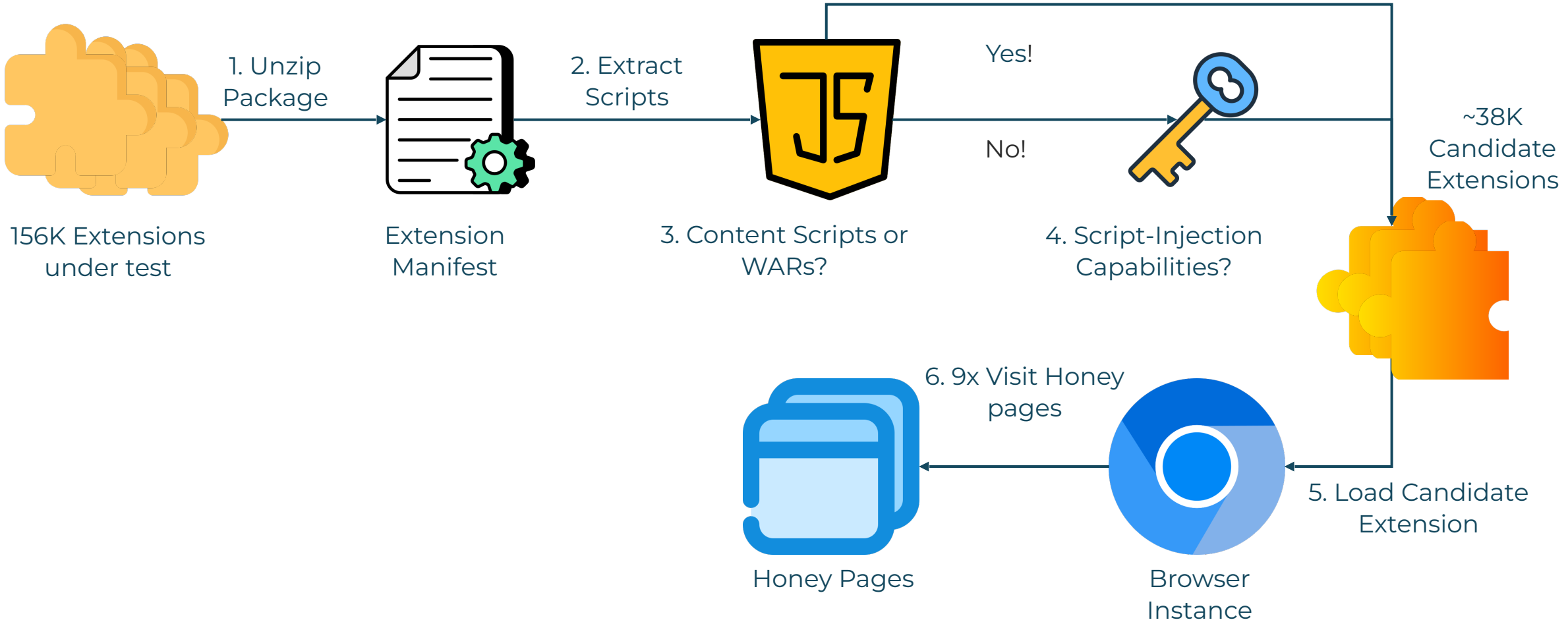
Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?



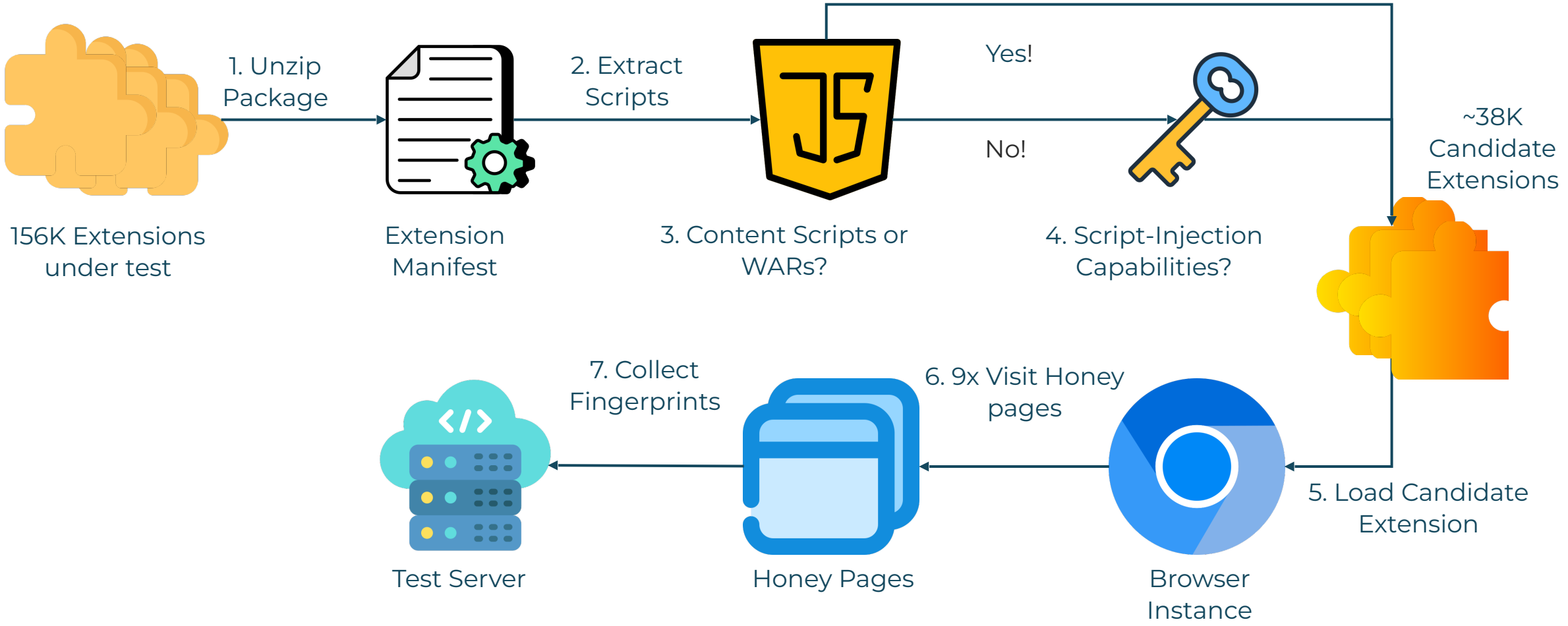
Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?



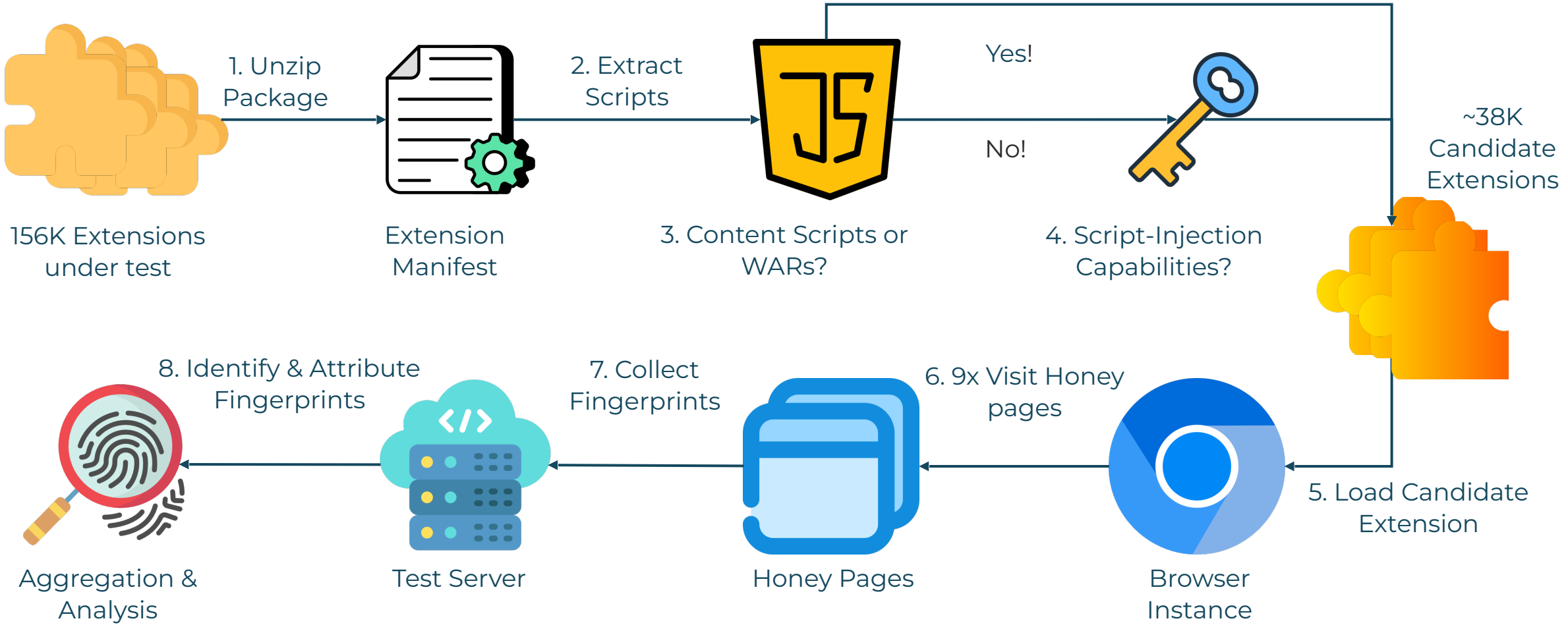
Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?



Raider: Pipeline

How many extensions can be uniquely fingerprinted through these observable side effects?



Evaluation I - *Chrome Extensions*

Evaluation I - *Chrome Extensions*

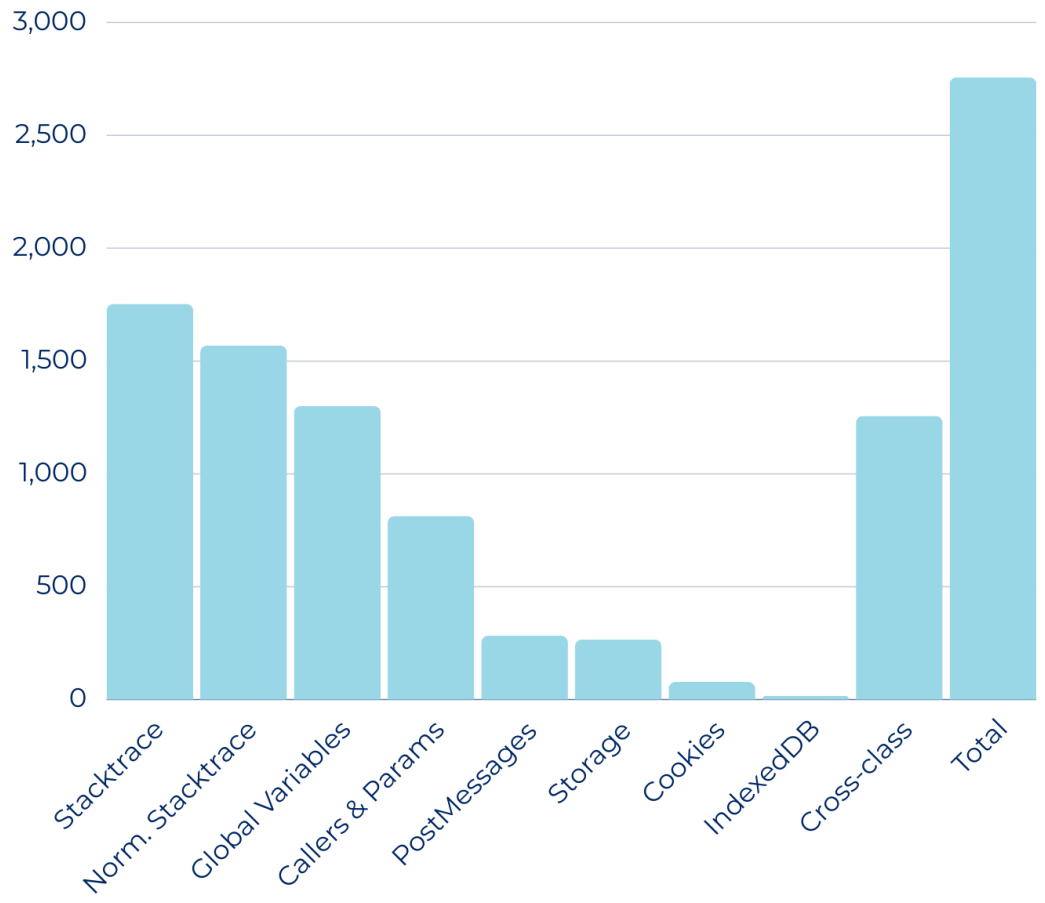
- Analyzed ~38K extensions.

Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.

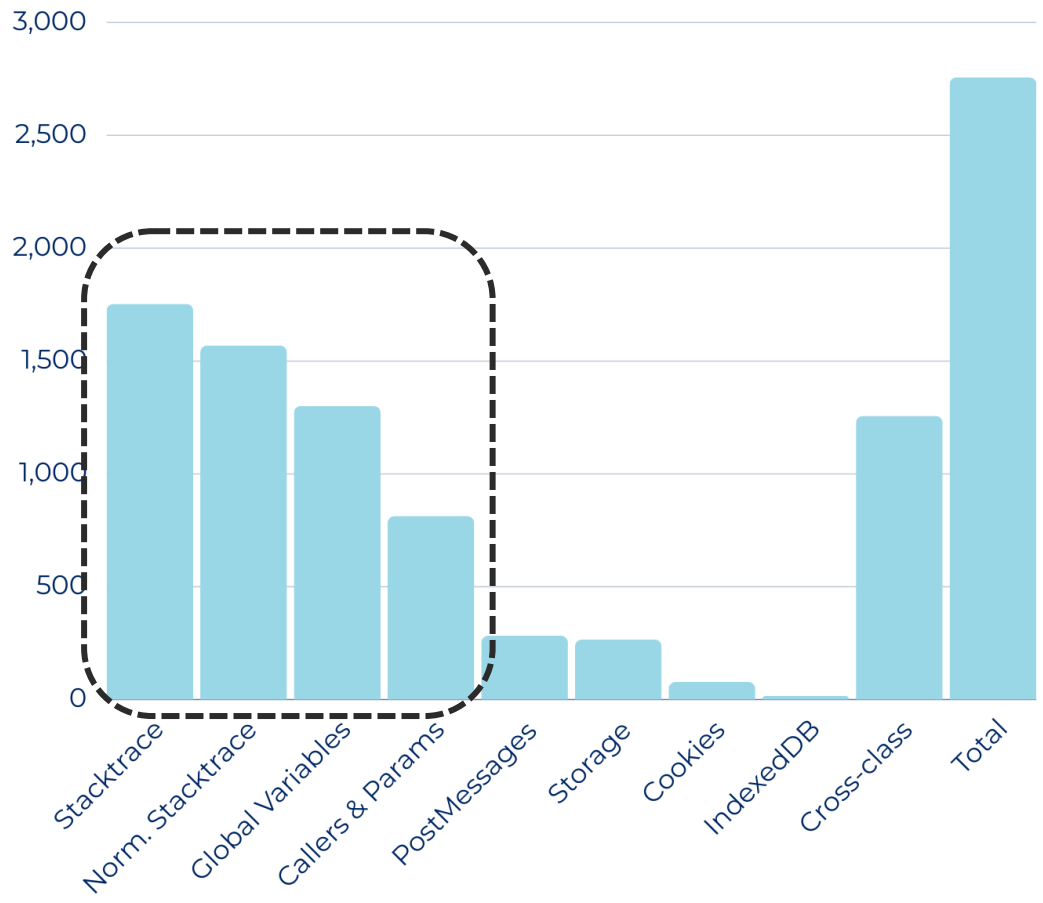
Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.



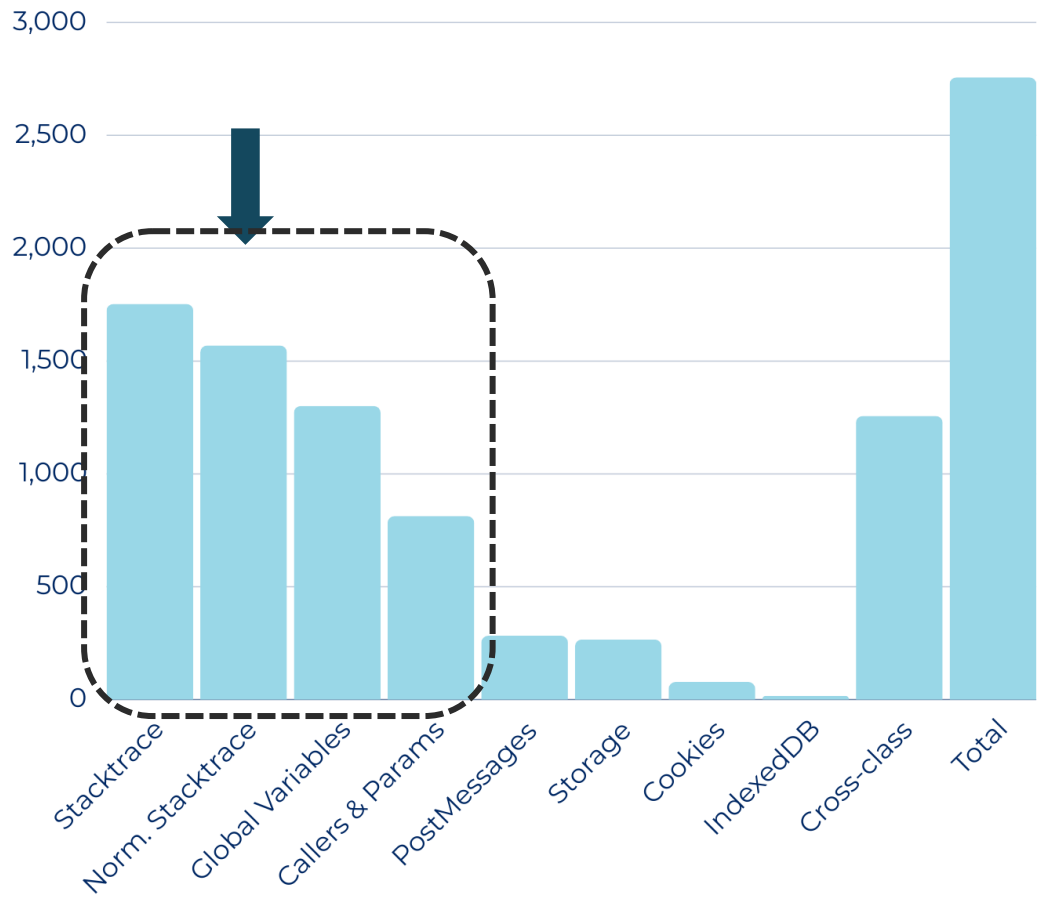
Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.
- Extensions often inject scripts into the global namespace.



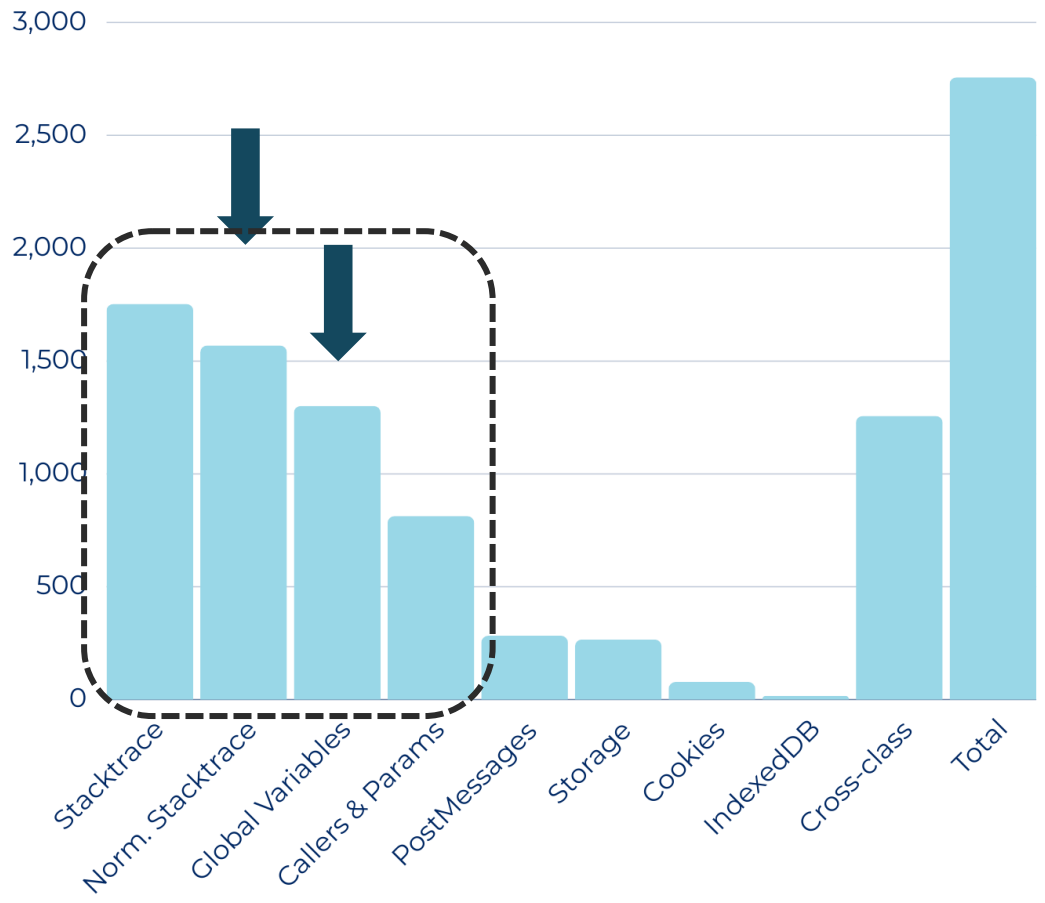
Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.
- Extensions often inject scripts into the global namespace.



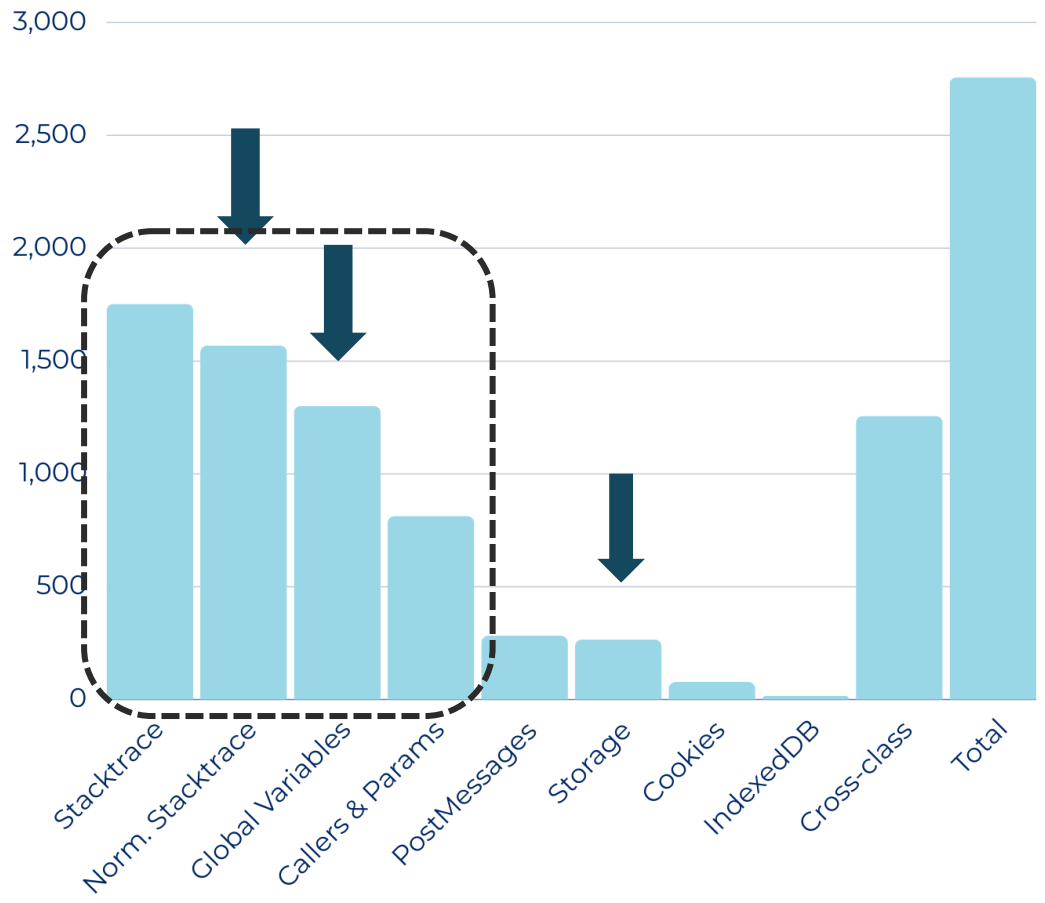
Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.
- Extensions often inject scripts into the global namespace.



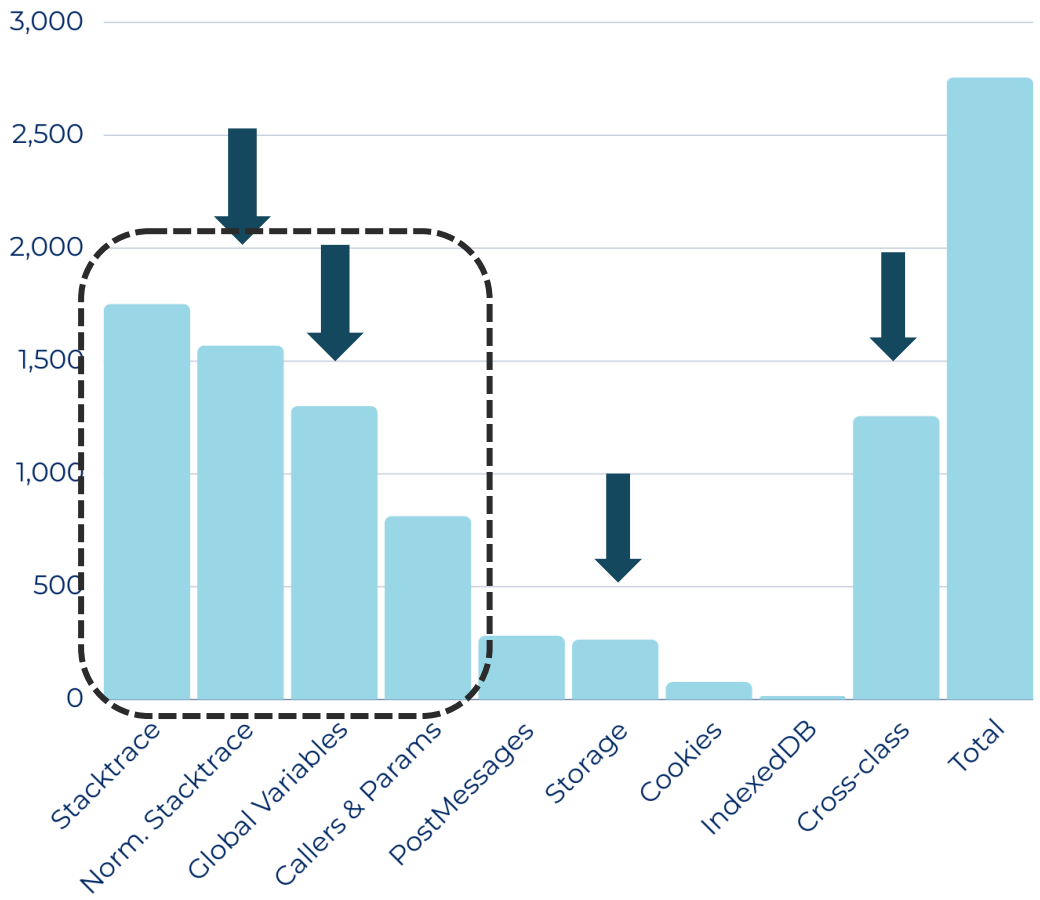
Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.
- Extensions often inject scripts into the global namespace.



Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.
- Extensions often inject scripts into the global namespace.
- Combination of multiple feature – yields fingerprints.



Evaluation II – *Multiple Extension Environment*

Evaluation II – *Multiple Extension Environment*

- Multiple extensions with conflicting/similar operations might impair fingerprinting capabilities.

Evaluation II – *Multiple Extension Environment*

- Multiple extensions with conflicting/similar operations might impair fingerprinting capabilities.
- Tests across different sample sizes of randomly sampled fingerprintable extensions.
  - *Sample Size (N)= 2, 3, 4, … 10.*
  - *True-positive rate average over five distinct runs.*

Evaluation II – *Multiple Extension Environment*

- Multiple extensions with conflicting/similar operations might impair fingerprinting capabilities.
- Tests across different sample sizes of randomly sampled fingerprintable extensions.
  - *Sample Size (N)= 2, 3, 4, … 10.*
  - *True-positive rate average over five distinct runs.*

Raider could still accurately fingerprint extensions with our newly discovered vectors (*Accuracy: 98%*)

Evaluation III - *Firefox Extensions*

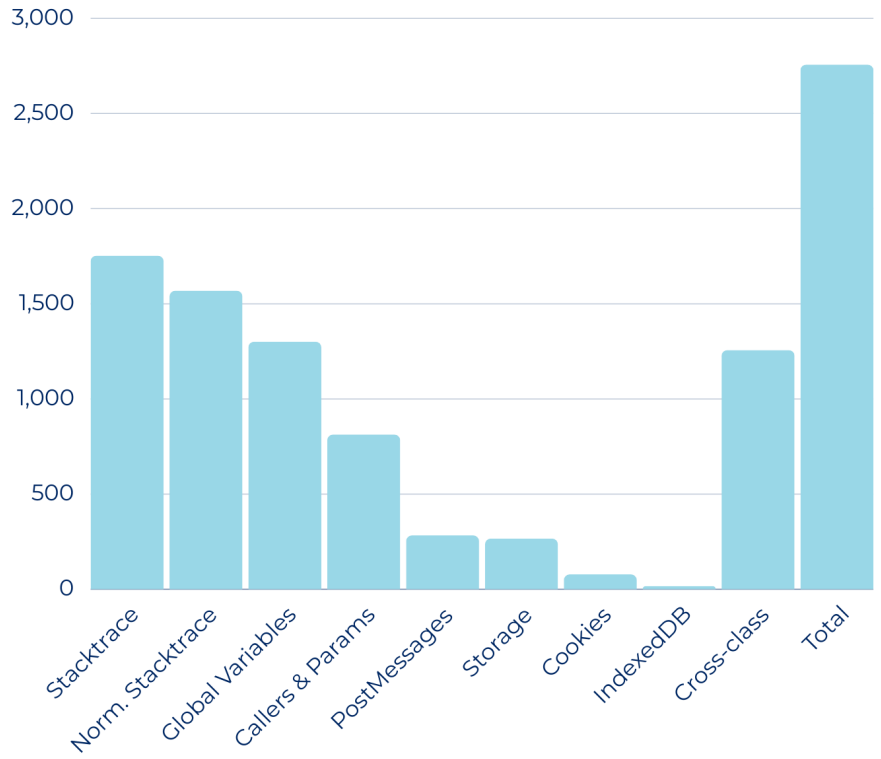
Evaluation III - *Firefox Extensions*

572 Firefox Extensions are also fingerprintable.

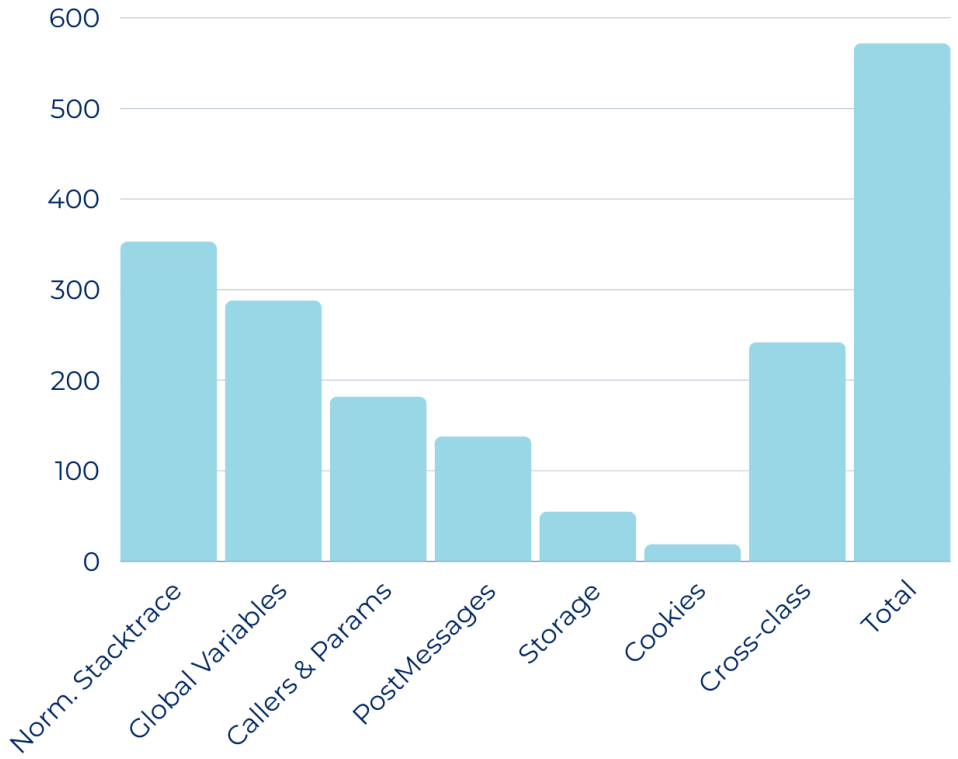
Evaluation III - *Firefox Extensions*

572 Firefox Extensions are also fingerprintable.

Raider (Chrome) Dataset



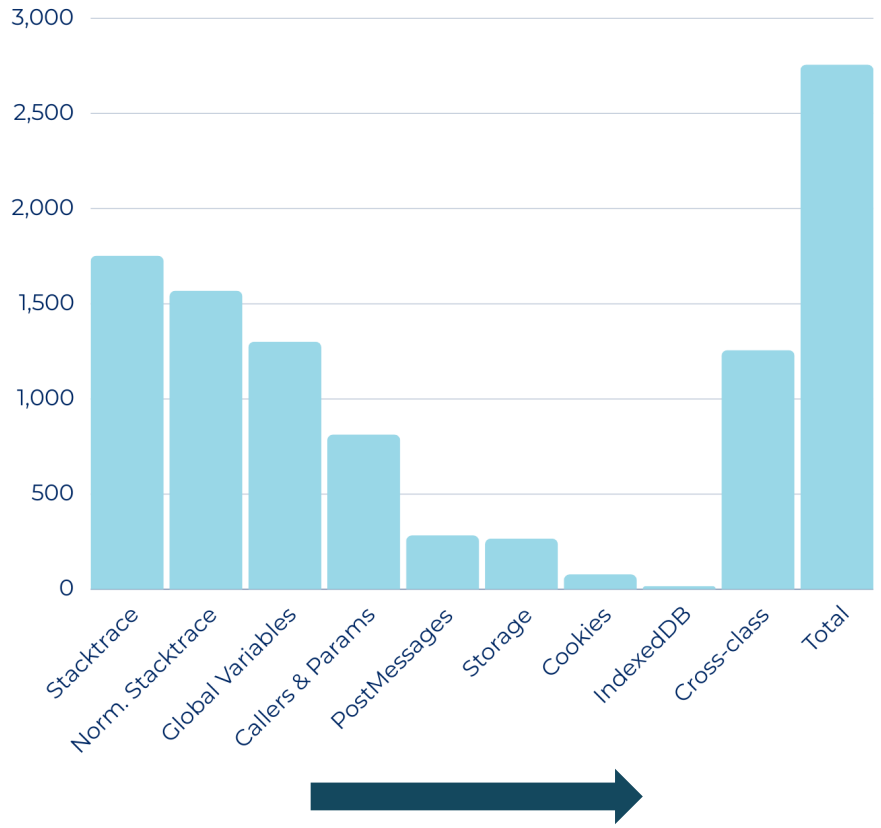
Firefox Dataset



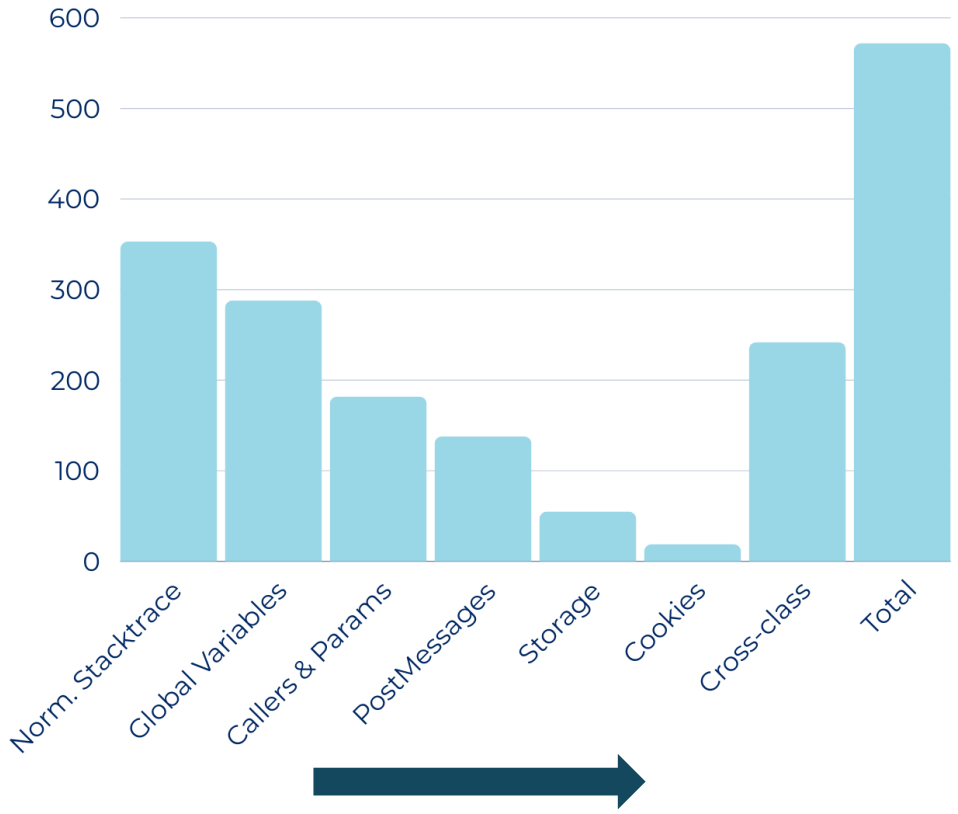
Evaluation III - *Firefox Extensions*

572 Firefox Extensions are also fingerprintable.

Raider (Chrome) Dataset



Firefox Dataset



Developers' Take

Developers' Take

We notified ~2K extension developers about the problem.

Developers' Take

We notified ~2K extension developers about the problem.

- Proof-of-concept testbed* for verification.

Test Page for Developers

Developers' Take

We notified ~2K extension developers about the problem.

- Proof-of-concept testbed* for verification.
- 16 of them positively acknowledged the problem.

Test Page for Developers

Developers' Take

We notified ~2K extension developers about the problem.

- Proof-of-concept testbed* for verification.
- 16 of them positively acknowledged the problem.
- They indicated the behavior (e.g., script injection) to be required for their extensions.

Test Page for Developers

Developers' Take

We notified ~2K extension developers about the problem.

- Proof-of-concept testbed* for verification.
- 16 of them positively acknowledged the problem.
- They indicated the behavior (e.g., script injection) to be required for their extensions.
  - At the moment, no mitigation strategies exist at browser level.

Test Page for Developers

Developers' Take

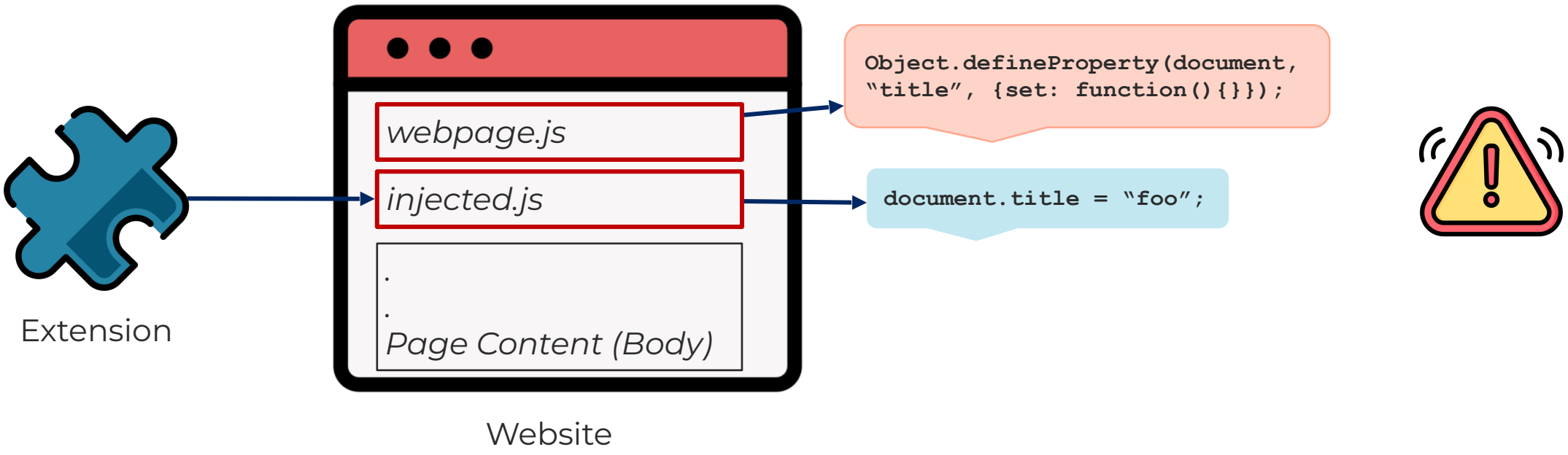
We notified ~2K extension developers about the problem.

- Proof-of-concept testbed* for verification.
- 16 of them positively acknowledged the problem.
- They indicated the behavior (e.g., script injection) to be required for their extensions.
  - At the moment, no mitigation strategies exist at browser level.
- 4 of them mentioned – *"...it should be the platform's responsibility to take care of such issues!"*.

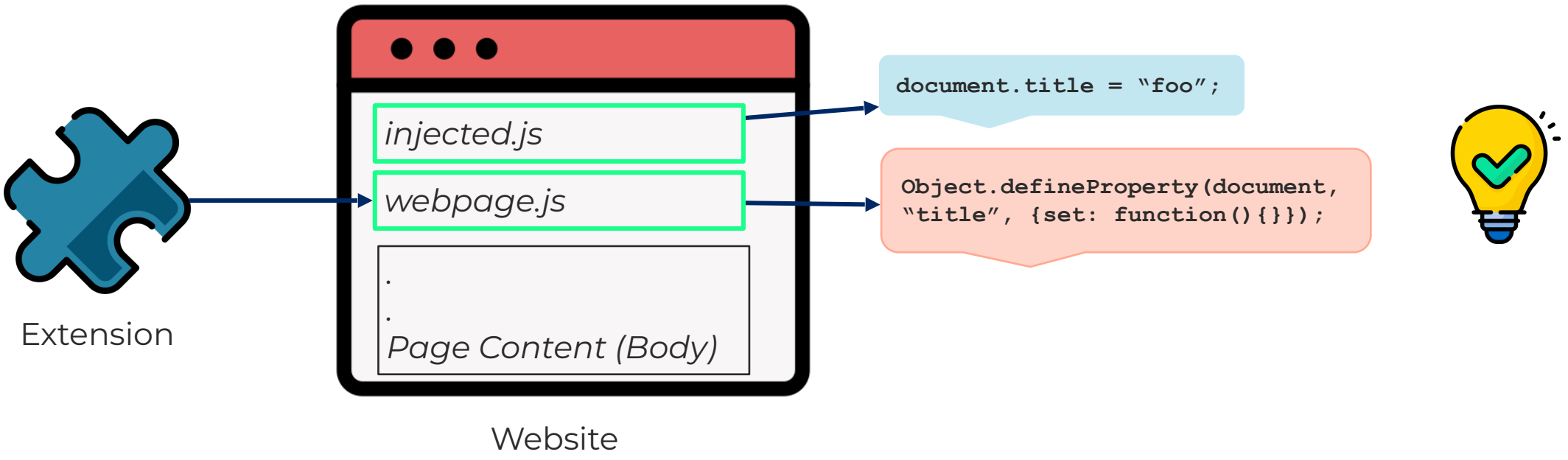
Test Page for Developers

Recommendations

Recommendations

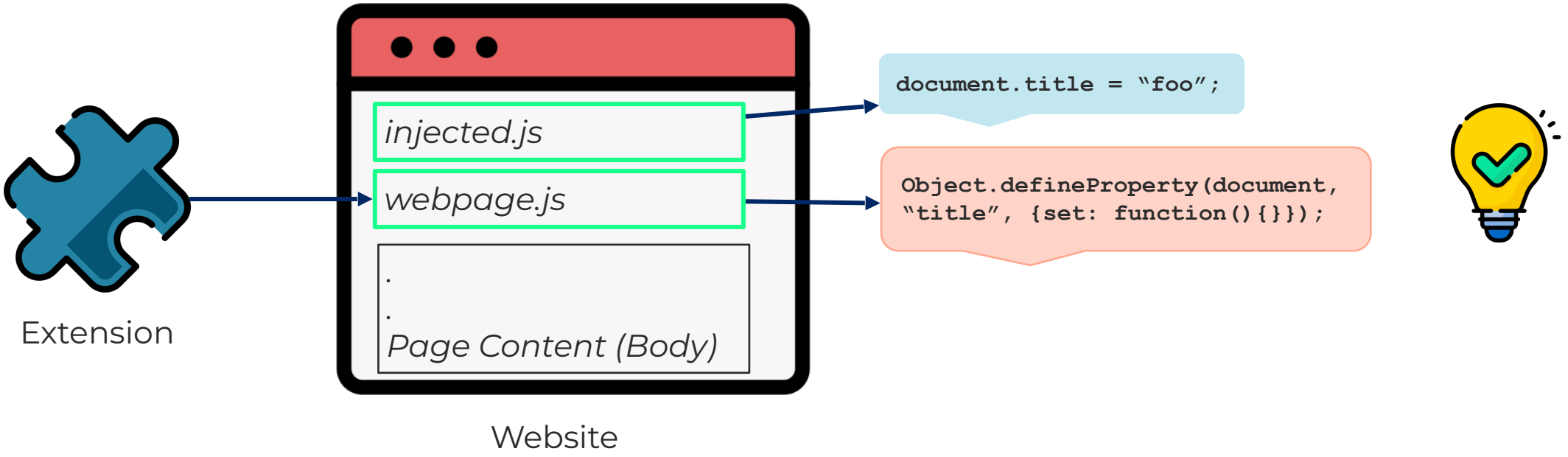


Recommendations



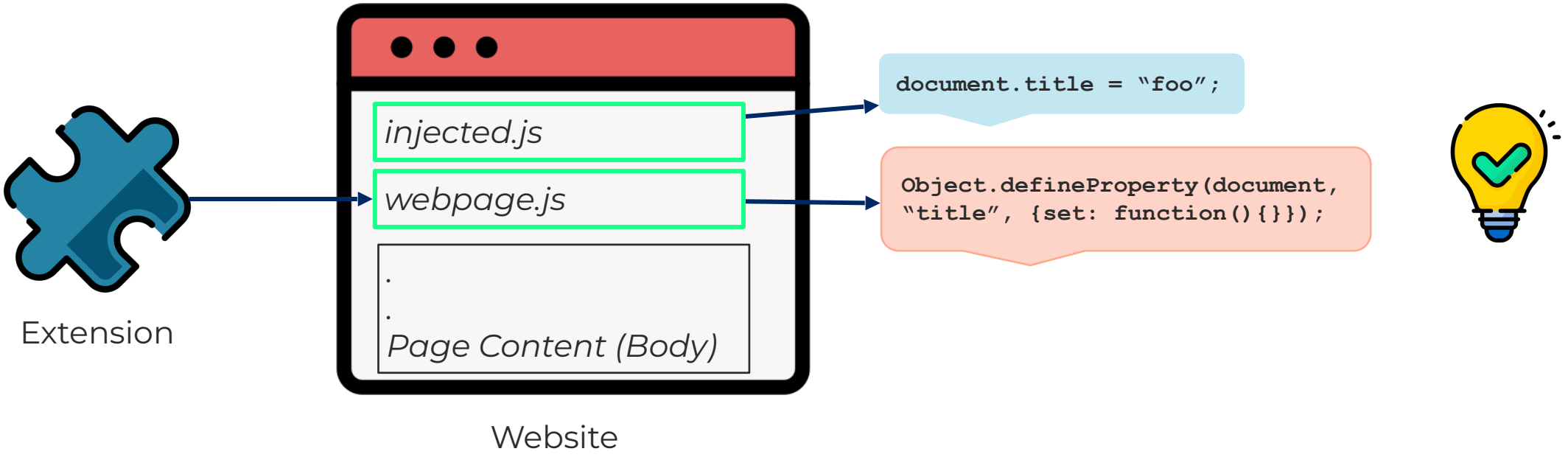
Recommendations

Extension-injected scripts must execute first



Recommendations

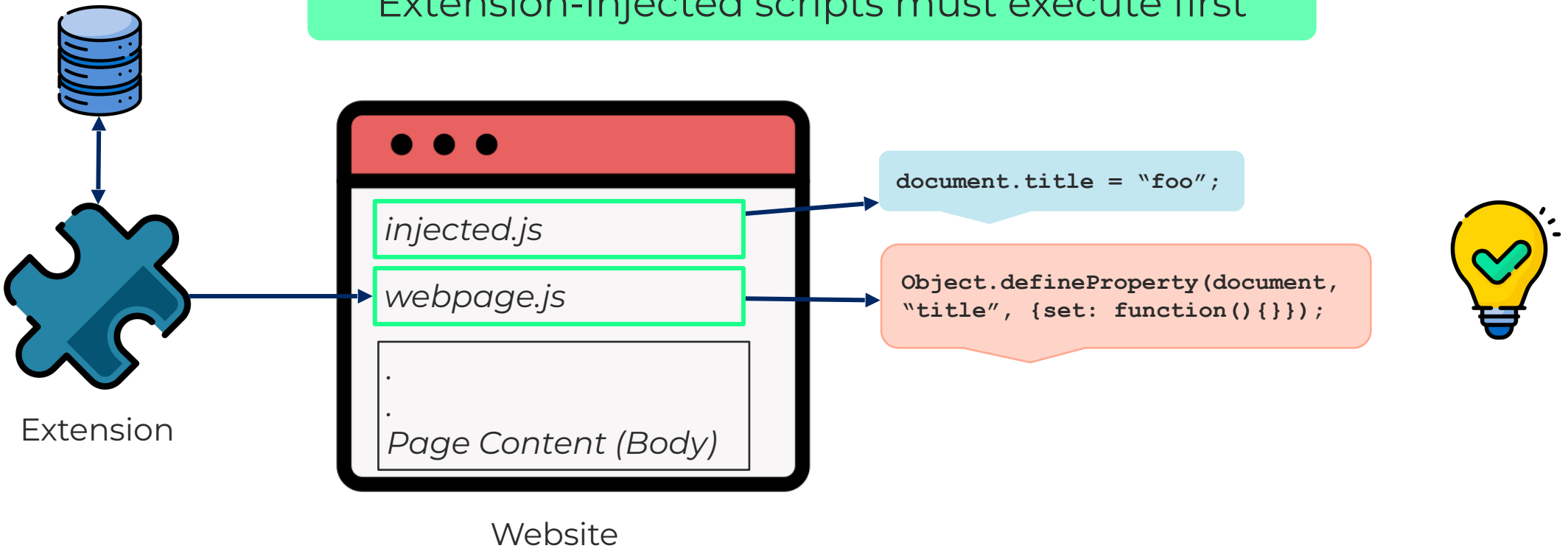
Extension-injected scripts must execute first



Immediately Invoked Function Expressions (*IIFEs*)

Recommendations

Extension-injected scripts must execute first



Immediately Invoked Function Expressions (*IIFEs*)

`chrome.storage` >>> Web Storage APIs

Summary

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.
- Modern browser architecture do not mitigate against these runtime behavior.

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.
- Modern browser architecture do not mitigate against these runtime behavior.
- We responsibly notified to the developers of affected extensions.

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.
- Modern browser architecture do not mitigate against these runtime behavior.
- We responsibly notified to the developers of affected extensions.
  - They heavily rely on the studied vectors for their extensions' functionality.

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.
- Modern browser architecture do not mitigate against these runtime behavior.
- We responsibly notified to the developers of affected extensions.
  - They heavily rely on the studied vectors for their extensions' functionality.
- We open-source our dataset and honey pages for developers and researchers.

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.
- Modern browser architecture do not mitigate against these runtime behavior.
- We responsibly notified to the developers of affected extensions.
  - They heavily rely on the studied vectors for their extensions' functionality.
- We open-source our dataset and honey pages for developers and researchers.

Test Page for Developers

More technical details in paper!

Summary

- We discover three new vectors to fingerprint browser extensions in the wild.
- Through our proposed vectors, we detected over 3K fingerprintable extensions across Web Stores.
- Modern browser architecture do not mitigate against these runtime behavior.
- We responsibly notified to the developers of affected extensions.
  - They heavily rely on the studied vectors for their extensions' functionality.
- We open-source our dataset and honey pages for developers and researchers.

Test Page for Developers

More technical details in paper!

shubh401/raider

THANK YOU!

@ap0ca1pse

Raider: Honey Pages

- Two distinct pages used for analysis
  - To collect execution traces from global namespace.
  - To collect data by polling storage APIs.
- Different test page features used during analysis.
  - Basic test page with minimal HTML structure.
  - Enhanced test page with broad range of HTML elements, also used by *Carnus*.
  - Also includes JavaScript for:
    - Mouse events (click, double click, etc.)
    - Keyboard events (hot keys, custom commands, etc.)
    - To model user-induced interaction/events.

Raider: API Hooks

- Overwrote many Global JavaScript APIs and properties to inject logger.
  - 571 Global JS APIs (e.g., `Array.prototype.from`).
  - 52 Global properties (e.g. `document.domain`)

```
1  function __hook(object, property, api) {
2    // Preserving native definition of the function.
3    let __originalFunc = object[property];
4    // Custom definition for Global APIs
5    function __customFunc() {
6      // Extracting API related information.
7      let context = this;
8      let args = Array(...arguments);
9      // Extracting the source code of the executing code.
10     let callerData = {};
11     let caller = arguments?.callee?.caller;
12     while (caller) {
13       callerName = caller.name;
14       callerFunc = caller.toString();
15       callerData[callerName] = callerFunc;
16       caller = caller?.arguments?.callee?.caller;
17     }
18     // Capturing the stack trace of the executing code.
19     let stacktrace = new Error().stack;
20     // Sending data to our test server.
21     logToServer({ api, context, args, stacktrace, callerData });
22     // Now, returning the result from executing native function.
23     return __originalFunc.apply(this, arguments);
24   }
25   // Replacing the native definition with custom definition.
26   object[property] = __customFunc;
27 }
28
29 //Instrumenting APIs now...
30 __hook(Array.prototype, "forEach", "Array.forEach");
```

Raider: Other Data Collection Techniques

```
// content_scripts.js
localStorage.setItem('foo', 'bar');
// attacker-webpage.js
for (let index = 0; index < localStorage.length; index++) {
  let key = localStorage.key(index);
  let value = localStorage.getItem(key);
  logStorage(key, value);
}
```

(a) *Storage* APIs scanning

```
// content_script.js
window.postMessage('Hello from CS!', '*');
// popup.js
window.addEventListener('message', function (event) {
  event.source.postMessage('Message received!');
});
// attacker-webpage.js
window.addEventListener('message', function (event) {
  logMessages(event.data);
});
```

(b) Intercepting postMessages

```
// injected-script.js
extension_key = "extension_value";
// attacker-webpage.js
for (let prop of Object.getOwnPropertyNames(window)) {
  logProperty(prop, window[prop]);
}
```

(c) Global variables set by extensions

Evaluation I - *Chrome Extensions*

- Analyzed ~38K extensions.
- 2,757 are fingerprintable.
- Extensions often inject scripts into the global namespace.
- `localStorage` for data storage – leads to observable existence.

Method	Usage	Repeated	Unique	Only	Installs
Global APIs	1,878	1,872	1,769	-	109,584,572
- Stacktrace	1,878	1,871	1,753	397	108,382,340
- Norm. Stacktrace	1,878	1,871	1,569	(237)	103,582,524
- Caller & Params	1,878	1,868	813	2	32,740,589
Variables	1,730	1,664	1,301	245	67,048,809
Cookies	201	198	154	78	4,709,235
Storage	634	623	391	266	8,317,933
IndexedDB	128	126	32	17	1,580,655
PostMessages	1,069	1,028	737	283	38,519,471
Cross-class	1,634	1,610	1,257	0	48,466,020
Total	3,398	3,308	2,747	-	169,093,032

Table 2: Results for the *Raider* dataset

Evaluation II – *Multiple Extension Environment*

- Multiple extensions with conflicting/similar operations might impair fingerprinting capabilities.
- Tested across different sample sizes of fingerprintable extensions.
  - *Sample Size (N)= 2, 3, 4, … 10.*
  - *True-positive rate average over five distinct runs.*
- *Raider* could still consistently fingerprint extensions with our newly discovered vectors (*98%*).

N	2	3	4	5	6	7	8	9	10	Avg.
TP (%)	99.7	99.4	99.3	98.8	97.5	96.6	96.4	97.5	97.4	98.0
FN (%)	0.3	0.6	0.7	1.2	2.5	3.4	3.6	2.5	2.6	1.9
FP (%)	0.4	0.4	0.4	0.4	0.4	0.4	0.5	0.4	0.5	0.4
F1 (%)	99.7	99.5	99.3	99.2	98.5	98.1	97.9	98.5	98.4	98.8

Table 3: Multi-extension results (average over five runs)

Evaluation III - *Firefox Extensions*

572 Firefox Extensions are also fingerprintable.

Raider Dataset

Method	Usage	Repeated	Unique	Only	Installs
Global APIs	1,878	1,872	1,769	-	109,584,572
- Stacktrace	1,878	1,871	1,753	397	108,382,340
- Norm. Stacktrace	1,878	1,871	1,569	(237)	103,582,524
- Caller & Params	1,878	1,868	813	2	32,740,589
Variables	1,730	1,664	1,301	245	67,048,809
Cookies	201	198	154	78	4,709,235
Storage	634	623	391	266	8,317,933
IndexedDB	128	126	32	17	1,580,655
PostMessages	1,069	1,028	737	283	38,519,471
Cross-class	1,634	1,610	1,257	0	48,466,020
Total	3,398	3,308	2,747	-	169,093,032

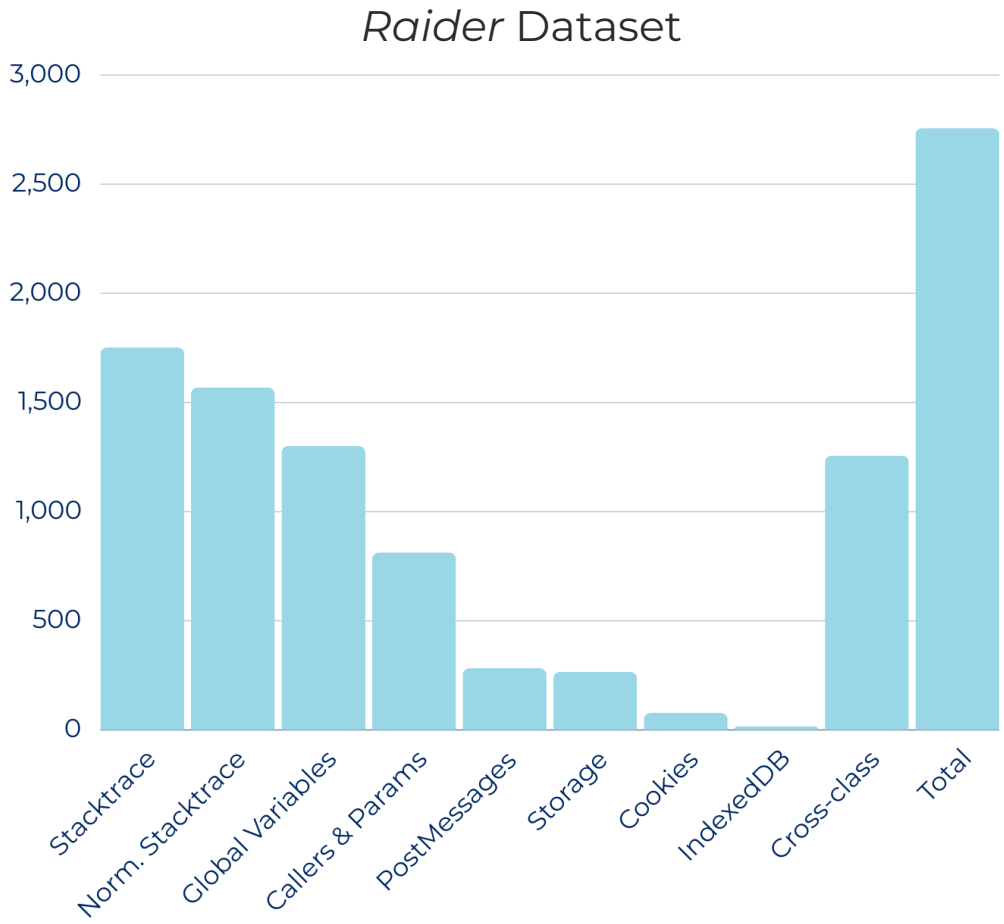
Table 2: Results for the *Raider* dataset

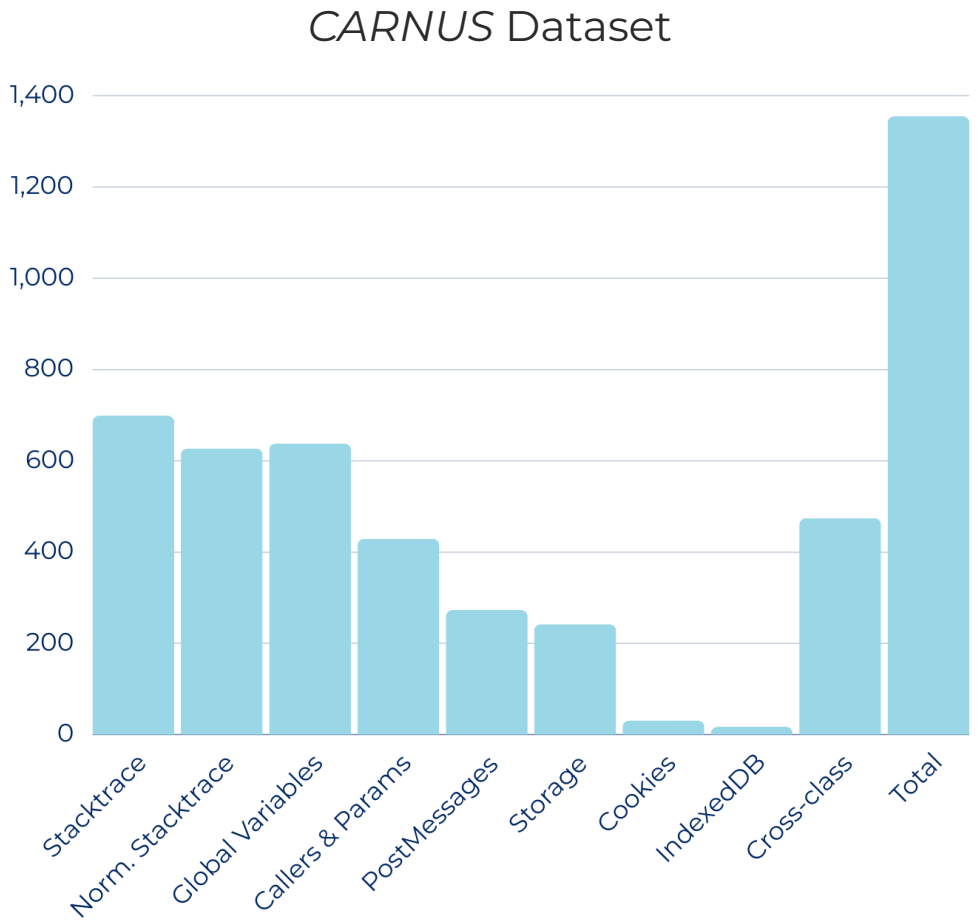
Firefox Dataset

Method	Usage	Repeated	Unique	Only
Global APIs	436	432	367	-
- Norm. Stacktrace	436	432	353	0
- Caller & Params	436	423	182	14
Variables	359	351	288	79
Cookies	25	24	19	14
Storage	85	84	55	43
IndexedDB	-	-	-	-
PostMessages	176	172	138	54
Cross-class	314	305	242	0
Total	689	682	572	-

Table 5: Results for the *Firefox* dataset

Evaluation IV – *Comparison with CARNUS*





Raider could fingerprint 484 extensions from CARNUS dataset not reported before.

Prior Studies on Extension Fingerprinting

- WAR-based Fingerprinting – through the static URL of included resources.
  - Dynamic Runtime Identifiers (`use_dynamic_uri`).
- DOM-based Fingerprinting – through transient or persistent DOM modification, internal or external communication channels.
  - Parallel DOM.
- Style-based Fingerprinting – style-based modifications.
  - ShadowDOM.
- User-induced side-effects – interaction-dependent behavior.
  - Code-level source validation for events (`event.isTrusted`).

S

whoami

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

German OWASP Day 2024

- PhD Student @CISPA Helmholtz Center for Information Security, Saarbrücken, DE.
- Likes to talk/hear on all things Applications Security & Data Privacy.
- Currently interested in Browser Extensions (and PhD memes 🫠).

Shubham Agarwal

Secure Web Applications Group