

Security & Privacy Self-Review Questionnaire — JS Self-Profiling Markers

Feature: ProfilerSample.marker (marker field added to the existing JS Self-Profiling API

<https://wicg.github.io/js-self-profiling/>)

Spec / proposal: <https://github.com/WICG/js-self-profiling/blob/main/markers.md>

Spec PR (moves IDL into the normative spec): <https://github.com/WICG/js-self-profiling/pull/89>

ChromeStatus: <https://chromestatus.com/feature/5201297767792640>

Prior security discussion: <https://github.com/WICG/js-self-profiling/pull/61>

Answers follow the questions in <https://www.w3.org/TR/security-privacy-questionnaire/>

Summary of the change under review. This feature adds one optional field, marker, to ProfilerSample. When the JS Self-Profiling sampling profiler takes a sample, marker labels what the user agent was doing at that instant: one of script, gc, style, layout, paint, or other. It adds no new *way to start profiling and no new data source* — *it annotates samples that the already-shipped Profiler API produces*. Exposure is conditioned on cross-origin isolation:

- Non-cross-origin-isolated documents: only style and layout markers are exposed; all others are reported as absent.
- Cross-origin-isolated documents (COOP+COEP): the full marker set is exposed.

Profiling itself must still be enabled by the site via the existing Document-Policy: js-profiling mechanism; markers do not change that gate.

01. What information does this feature expose, and for what purposes?

It exposes a coarse category label describing which internal user-agent activity was on the stack when a sample was taken (script / gc / style / layout / paint / other). The purpose is to let a page attribute time in its own self-profiling traces to non-JavaScript browser work (e.g. layout or garbage collection), which otherwise appears as unexplained gaps. All information stays within the same-origin document that started the profiler; nothing is sent cross-origin and nothing new is disclosed to other parties.

02. Do features in your specification expose the minimum amount of information necessary?

Yes. The marker is a single enumerated category, not a duration, address, memory figure, or internal detail. In non-isolated documents the set is further reduced to style and layout only — the subset whose timing a page can already observe same-origin — and the potentially more sensitive markers (gc, paint, script) are withheld unless the document is cross-origin isolated.

03. Do the features expose personal information, PII, or information derived from either?

No. Markers describe user-agent engine activity, not the user. They contain no personal or personally identifiable information and are not derived from any.

04. How do the features deal with sensitive information?

The feature does not handle sensitive user information. The one sensitivity *class it touches is timing* — *fine-grained timing of browser internals can be a side channel*. This is addressed by gating: markers that have no same-origin observable equivalent (gc, paint, script) are exposed only in cross-origin-isolated contexts, matching the isolation boundary already used for high-resolution and cross-origin-sensitive timing. The style/layout subset exposed to non-isolated documents reflects timing a page can already measure via forced synchronous reflow (Element.getBoundingClientRect() forces layout; getComputedStyle() forces style/layout).

05. Do the features expose related but distinct information that may not be obvious to users?

No. The marker is exactly what it appears to be — a category for a sample. It carries no hidden secondary signal. Any coarse timing inference an attacker could draw from marker frequency in a non-isolated context is already available today through the forced-reflow APIs noted above, so no new distinct information is introduced.

06. Do the features introduce state that persists across browsing sessions?

No. Marker data exists only within the in-memory trace of a single Profiler instance and is discarded when the profiler is stopped and the trace is garbage collected. Nothing is stored or persisted.

07. Do the features expose information about the underlying platform to origins?

Only indirectly and coarsely: the mix of markers reflects that a given engine performs style, layout, paint, and garbage collection — which is already true of and observable in every browser. No hardware, OS, build, or configuration detail *is exposed*. *Sample timing is subject to the cross-origin-isolation gating* described above.

08. Does this specification allow an origin to send data to the underlying platform?

No.

09. Do features in this specification enable access to device sensors?

No.

10. Do features in this specification enable new script execution or loading mechanisms?

No.

11. Do features in this specification allow an origin to access other devices?

No.

12. Do features in this specification allow an origin some measure of control over a user agent's native UI?

No.

13. What temporary identifiers do the features create or expose to the web?

None. Markers are fixed enumerated categories, not identifiers, and are not unique to a user, session, or device.

14. How does this specification distinguish between behavior in first-party and third-party contexts?

Profiling is controlled by the existing Document-Policy: js-profiling mechanism, which a document (including a subframe) must opt into. The marker addition inherits that model and does not introduce first-/third-party-specific behavior of its own. The full marker set additionally requires cross-origin isolation of the document.

15. How do the features work in the context of a browser's Private Browsing or Incognito mode?

Identically. The feature has no persistent state and no cross-mode signal, so it behaves the same in normal and private/incognito browsing and provides no way to distinguish them.

16. Does this specification have both "Security Considerations" and "Privacy Considerations" sections?

The base JS Self-Profiling specification includes security/privacy discussion. The markers proposal (markers.md) documents the cross-origin-isolation gating and the rationale for the exposed non-isolated subset; this content is being folded into the normative spec via WICG/js-self-profiling#89, and the security analysis is tracked in WICG/js-self-profiling#61.

17. Do features in this specification enable origins to downgrade default security protections?

No. The feature only tightens with isolation state — non-isolated documents receive strictly less (only style/layout). It provides no mechanism to weaken any existing protection.

18. What happens when a document using your feature is kept alive in BFCache after navigation and potentially reused?

No special interaction. Marker data lives only inside an active Profiler instance's in-memory trace and carries no cross-navigation state, so BFCache storage and reuse neither expose nor accumulate any additional information.

19. What happens when a document that uses your feature gets disconnected?

The in-memory trace becomes unreachable and is garbage collected with the document. No state survives disconnection.

20. Does your spec define when and how new kinds of errors should be raised?

No new error types. The marker is simply present or absent on a sample; absence (including the filtered-out markers in non-isolated documents) is represented by the field being unset rather than by an error.

21. Does your feature allow sites to learn about the user's use of assistive technology?

No. Marker categories reflect engine work (script/gc/style/layout/paint) and are independent of whether assistive technology is in use.

22. What should this questionnaire have asked?

*The relevant axis for this feature is *timing side channels from sampled engine state*, which the questionnaire covers under sensitive-information/timing. The questionnaire adequately surfaces the one real consideration here (timing), which we mitigate via the cross-origin-isolation gate plus the already-observable style/layout subset for non-isolated documents.*

Notes for reviewers (changes since prior review)

- The cross-origin-isolated full marker set was security-reviewed and approved in 2021 (<https://github.com/WICG/js-self-profiling/pull/61>; Document-Policy / security-bit change). No change to that behavior.
- The **new delta** under review is exposing only style and layout markers in **non-cross-origin-isolated** documents. Rationale: that timing is already same-origin observable via forced synchronous reflow, so no new information class is introduced; the markers provide the same signal sampled rather than forced (strictly less capability). All markers with no same-origin equivalent remain cross-origin-isolated-gated.