Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 8, 2019

M. West
Google

May 7, 2019

First-Party Sets and SameSite Cookies
draft-west-cookie-samesite-firstparty-00

Abstract

This document proposes the addition of two new values to the "SameSite" cookie attribute defined in RFC6265bis [I-D.ietf-httpbis-rfc6265bis]: "FirstPartyLax" and "FirstPartyStrict". These values are conceptually similar to the existing "Lax" and "Strict" values, but base the delivery checks on the First-Party Sets [first-party-set] of a request's initiator and target, rather than on their respective registrable domains. This widens the scope of a given cookie's applicability, enabling entities that have sharded themselves across multiple registrable domains to maintain HTTP state without exposing themselves to the risks of "SameSite=None".

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 November 8, 2019.

Table of Contents

1. Introduction

The "SameSite" attribute enables developers to limit the scope of a given cookie's delivery, mitigating the risks of some classes of cross-site request forgery (CSRF) attack by preventing certain cookies from being delivered along with requests that are initiated from a cross-site context.

For example, consider the exciting and dynamic "https://internet-bookstore.example/", which uses "SameSite=Lax" cookies as one layer in its defense against CSRF attack. If "https://example.com" includes resources from "https://internet-bookstore.example/", the request will be considered cross-site, and the authentication cookies will not be delivered. Without that state, CSRF attacks will be significantly less effective.

When the site expands into new locations, it may wish to register a domain under a localized TLD, perhaps "https://internet-bookstore.测试/". Likewise, it may decide to shard itself into distinct brands, like "https://internet-things-other-than-books-store.example/". Though the same entity controls each of these origins, they have distict registrable domains, and therefore the authentication cookie noted above will not be delivered from one

site to resources on another. This frustrates a number of reasonable use cases, including single-sign on. Today, "SameSite=None" is necessary in order to support these use cases by enabling a given cookie to be delivered across registrable domains. "SameSite=None", unfortunately, exposes the site to more risk than it would prefer, as it removes a layer of CSRF defense.

First-Party Sets [first-party-set] proposes a mechanism by which developers can bind each of their distinct registrable domains into a set which mutually agrees to be treated as a single entity. It would be helpful if this concept could be folded into the "SameSite" attribute, perhaps via new "FirstPartyLax" and "FirstPartyStrict" values. These could be conceptually similar to the existing "Lax" and "Strict" values, but base their delivery checks on the First-Party Sets of a given request's initiator and target, rather than on their respective registrable domains.

This document spells out that proposal in a bit more detail.

2. Conventions and Definitions

2.1. Conformance

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Syntax

This document adjusts some syntax from [I-D.ietf-httpbis-rfc6265bis], and in doing so, relies upon the Augmented Backus-Naur Form (ABNF) notation of [RFC5234].

2.3. Terms

HTTP requests are considered "same-site" or "cross-site", as defined in [I-D.ietf-httpbis-rfc6265bis].

First-Party Sets are defined in [first-party-set]. Two origins ("A" and "B") are said to be in the same first-party set if the first-party set associated with "A" contains "B".

A request is considered to be "first-party" if the target origin is in the same first-party set as the request's initiator, and "third-party" otherwise. That is, for a given request ("r"), the following algorithm returns "first-party" or "third-party":

1. If "request" is "same-site", return "first-party".

2. Let "target" be "r"'s current URL's origin.

3. If "target" is in the same first-party set as "r"'s client's origin, return "first-party".

4. Return "third-party".

A document is considered "first-party with its ancestors" if its origin is in the same First-Party Set with the origins of each of the document's ancestors [HTML]. That is, for a given document ("d"), the following algorithm returns "first-party" or "third-party":

1. If "d"'s browsing context is a top-level browsing context, return "first-party".

2. Let "set" be "d"'s origin's First-Party Set.

3. For each "ancestor" in "d"'s browsing context's ancestor browsing contexts:

    1. If "ancestor"'s active document's origin is not contained within "set", return "third-party".

4. Return "first-party".

ISSUE: Move these definitions to the First-Party Sets spec, when one exists.

3. The "FirstParty" value of the "SameSite" attribute

[I-D.ietf-httpbis-rfc6265bis] defines three values for cookies' "SameSite" attribute: "None", which enables delivery for same-site and cross-site requests; "Strict", which enables delivery only for same-site requests; and "Lax", which enables delivery for same-site requests as well as for cross-site top-level navigations.

In the presence of first-party sets, it makes sense to extend this syntax a bit to include "FirstParty", which will allow delivery of cookies within a first-party set, and therefore will support the use cases that first-party sets addresses (a given first-party's single sign-on, for instance). For example, given two distinct origins "https://sso.example/" and "https://application.example/" that are contained in the same first-party set:

- Requests from "https://application.example/" to "https://sso.example/" may not contain any cookies set with

"SameSite=Lax" or "SameSite=Strict", but only those set as "SameSite=None".

- Requests from "https://application.example/" to "https://sso.example/" may contain any cookies set with "SameSite=FirstParty" or "SameSite=None".

To implement this change, adjust [I-D.ietf-httpbis-rfc6265bis] as follows:

First, change the "samesite-value" definition from:

```
samesite-value    = "Strict" / "Lax" / "None"
```

to:

```
samesite-value    = "Lax" / "Strict" / "FirstPartyLax" / "FirstPartyStrict" /
"None"
```

Second, alter the "SameSite" attribute's processing algorithm (Section 5.3.7 of [I-D.ietf-httpbis-rfc6265bis]) to add a new step 4 and 5:

4. If cookie-av's attribute-value is a case-insensitive match for "FirstPartyLax", set `enforcement` to "FirstPartyLax".

5. If cookie-av's attribute-value is a case-insensitive match for "FirstPartyStrict", set `enforcement` to "FirstPartyStrict".

Third, alter the cookie storage model (Section 5.4 of [I-D.ietf-httpbis-rfc6265bis]) as follows:

Change step 14.1 from:

1. If the cookie was received from a "non-HTTP" API, and the API was called from a context whose "site for cookies" is not an exact match for request-uri's host's registered domain, then abort these steps and ignore the newly created cookie entirely.

to:

1. If the cookie was received from a "non-HTTP" API:

    1. If the cookie's `same-site-flag` is "Lax" or "Strict", and the API was called from a context whose "site for cookies" is not an exact match for request-uri's host's registered domain, then abort these steps and ignore the newly-created cookie entirely.

    2. If the cookie's same-site-flag` is "FirstPartyLax" or "FirstPartyStrict", and the API was called from a context that is not first-party with its ancestors, then abort these steps and ignore the newly-created cookie entirely.

Change step 14.2 from:

2. If the cookie was received from a "same-site" request, skip the remaining substeps and continue processing the cookie.

to:

2. If the cookie's `same-site-flag` is "Lax" or "Strict", and the cookie was received from a "same-site" request, then skip the remaining substeps and continue processing the cookie.

Add a new step 14.3 after the new step 14.2:

3. If the cookie's `same-site-flag` is "FirstPartyLax" or "FirstPartyStrict", and the cookie was received from a "first-party" request, then skip the remaining substeps and continue processing the cookie.

Fourth, alter the last conditional in step 1 if the "Cookie" header algorithm (Section 5.5 of [I-D.ietf-httpbis-rfc6265bis]) from:

* If the cookie's same-site-flag is not "None", and the HTTP request is cross-site then exclude the cookie unless all of the following statements hold:

    1. The same-site-flag is "Lax".

    2. The HTTP request's method is "safe".

    3. The HTTP request's target browsing context is a top-level browsing context.

to:

* If the HTTP request is cross-site, then exclude the cookie unless one of the following statements holds:

    1. The cookie's `same-site-flag` is "None".

    2. The cookie's `same-site-flag` is either "Lax" or "FirstPartyLax", the HTTP request's method is "safe", and the HTTP request's target browsing context is a top-level browsing context.

    3. The cookie's `same-site-flag` is either "FirstPartyLax" or "FirstPartyStrict", and the HTTP request is a first-party request.

4. Security and Privacy Considerations

4.1. CSRF

Both "FirstPartyLax" and "FirstPartyStrict" provide weaker defenses against CSRF than their "Lax" and "Strict" counterparts, as they enable authenticated requests from a larger set of initiating contexts. That said, they also provide deployment benefits, as they're usable in some contexts where "Lax" and "Strict" would be too restrictive (e.g. the localized registrable domains in the introduction).

4.2. Secure Transport

First-Party Sets can only be created for secure origins, as unauthenticated transport doesn't give any guarantees that the assertions we use to build the set are in fact being delivered by the entity which controls the server. This has the side-effect of ensuring that "FirstPartyLax" and "FirstPartyStrict" cookies can only be delivered to secure cross-site origins, which has the exciting side effect of providing limited mitigation of monitoring by network attackers [RFC7258].

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[first-party-set]
West, M., "First-Party Sets", n.d., <https://mikewest.github.io/first-party-sets/>.

[I-D.ietf-httpbis-rfc6265bis]
Barth, A. and M. West, "Cookies: HTTP State Management Mechanism", draft-ietf-httpbis-rfc6265bis-03 (work in progress), April 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

6.2. Informative References

[HTML] Hickson, I., Pieters, S., van Kesteren, A., Jaegenstedt, P., and D. Denicola, "HTML", n.d., <https://html.spec.whatwg.org/>.

[I-D.west-http-state-tokens]
West, M., "HTTP State Tokens", draft-west-http-state-tokens-00 (work in progress), March 2019.

[mixed-content]
West, M., "Mixed Content", n.d., <https://w3c.github.io/webappsec-mixed-content/>.

[pref-cookie]
Soltani, A., Peterson, A., and B. Gellman, "NSA uses Google cookies to pinpoint targets for hacking", December 2013, <https://www.washingtonpost.com/news/the-switch/wp/2013/12/10/nsa-uses-google-cookies-to-pinpoint-targets-for-hacking/>.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <https://www.rfc-editor.org/info/rfc6265>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

Acknowledgments

Conversations with a number of folks at 2019's HTTP Workshop helped me clarify my thinking around the incremental improvements we can make to cookies. In particular, Martin Thomson and Anne van Kesteren provided insightful feedback.

Author's Address

Mike West
Google

Email: mkwst@google.com
URI: https://www.mikewest.org/