Workgroup: HTTP
Internet-Draft:
draft-cutler-httpbis-partitioned-cookies-01
Updates: 6265 (if approved)
Published: 10 November 2022
Intended Status: Informational
Expires: 14 May 2023
Authors: D. Cutler
Google LLC

Cookies Having Independent Partitioned State specification

Abstract

This document updates RFC6265bis, defining a new attribute, Partitioned, which restricts the contexts in which a cookie is available to only those whose top-level document is same-site with the top-level document that initiated the request that created the cookie. These cookies are referred to as "partitioned cookies" and allow embedded sites which are cross-site with the top-level frame to have access to HTTP state which cannot be used for tracking across multiple top-level sites.

About This Document

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

The latest revision of this draft can be found at https://DCtheTall.github.io/CHIPS-spec/draft-cutler-httpbis-partitioned-cookies.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-cutler-httpbis-partitioned-cookies/.

Discussion of this document takes place on the HTTP Working Group mailing list (mailto:ietf-http-wg@w3.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/DCtheTall/CHIPS-spec.

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 14 May 2023.

Table of Contents

1. Introduction
2. Conventions and Definitions
  2.1. The Partitioned attribute
  2.2. Computing the cookie partition key
  2.3. Using Set-Cookie with Partitioned
  2.4. Partitioned Cookies with the Same Name/Domain/Path
  2.5. Attaching a Partitioned Cookie to a Request
3. Security Considerations
  3.1. Partitioned requires Secure
  3.2. Partitioned cookies and XSS attacks
  3.3. Partitioned cookies and CSRF attacks
  3.4. State proliferation for denial of service
  3.5. Partitioned cookies improve user privacy
  3.6. Avoiding cross-partition leaks
4. Implementation Considerations
  4.1. Applying Limits to Partitioned Cookie Jars
  4.2. Third-Party Cookie Controls
  4.3. Partitioned Cookies and Clear-Site-Data
5. IANA Considerations
6. References
  6.1. Normative References
  6.2. Informative References
Acknowledgments
Author's Address

1. Introduction

In order to increase privacy on the web, browser vendors are either planning or already shipping restrictions on cross-site tracking. This includes phasing out support for third-party cookies, as defined in Section 7.1 of [RFC6265bis].

Although third-party cookies can enable third-party sites to track user behavior across different top-level sites, there are some cookie use cases on the web today where cross-domain subresources require some notion of session or persistent state that is scoped to a user's activity on a single top-level site.

In order to meet these use cases, this document proposes changes to RFC6265bis to support a new cookie attribute, Partitioned, which restricts the contexts that a cookie is available to only those whose top-level document is same-site with the top-level document that the cookie was created in. This attribute will allow embedded sites to use HTTP state without giving them the capability to link user behavior across distinct top-level sites.

2. Conventions and Definitions

2.1. The Partitioned attribute

Below is the definition of the Partitioned attribute. This could be added as a new subsection of Section 4.1.2 (Semantics (Non-Normative)) of [RFC6265bis]:

> The Partitioned attribute limits the scope of the cookie such that it will only be sent when the site of the top-document (defined in section 5.2) is same-site with the top-document when the > cookie was created. Cookies set with this attribute are referred to as "partitioned cookies".

2.2. Computing the cookie partition key

Below is the algorithm that browsers can use to compute a request's cookie partition key. This algorithm could be added after Section 5.2 ("Same-site" and "cross-site" Requests) of [RFC6265bis]:

> 1. Let top-document be the active document in document's browsing context's top-level browsing context.
>
> 2. Let "cookie-partition-key" be the site of the top-document when the user agent made the request.
>
> 3. If the cookie is being read or written via a "non-HTTP" API, then cookie-partition-key is the site (as defined in [HTML]) of the top-document of the document associated with the non-HTTP API.

2.3. Using Set-Cookie with Partitioned

Below is the algorithm that browsers can use to parse cookie lines with this attribute. This algorithm could be added as a new subsection of Section 5.4 (The Set-Cookie Header Field) of [RFC6265bis]:

> If an attribute-name case-insensitively matches the string "Partitioned" then the user-agent MUST append an attribute to the

> cookie-attribute-list with an attribute-name of "Partitioned" and an empty attribute value.

We could add an attribute to the cookie storage model described in the first paragraph of 5.5 (Storage Model) to include a new attribute on each cookie called the partition-key (to differentiate it from cookie-partition-key defined in a prior section). The following could also be added as an additional step to section 5.4:

> 1. If the cookie-attribute-list does not contain an attribute with an attribute-name of "Partitioned", set partition-key to null.
>
>   If the cookie-attribute-list does contain an attribute with an attribute-name of "Partitioned" and the secure-only-flag is false, abort these steps and ignore the cookie entirely.
>
>   Otherwise, set partition-key to cookie-partition-key defined in section 5.X.X.

2.4. Partitioned Cookies with the Same Name/Domain/Path

In order to prevent cross-partition leaks, we need to allow sites to set cookies with the same name, domain, and path as long as they have different partition keys. In order to achieve this, we suggest the following edit to step 22 of Section 5.5 (Storage Model) of [RFC6265bis], note that steps b-d below are not changed.

> 1. If the cookie store contains a cookie with the same name, domain, host-only-flag, path, and partition-key as the newly-created cookie:
>
>   a. Let old-cookie be the existing cookie with the same name, domain, host-only-flag, path, and partition-key as the newly-created cookie. (Notice that this algorithm maintains the invariant that there is at most one such cookie.)
>
>   b. If the newly-created cookie was received from a "non-HTTP" API and the old-cookie's http-only-flag is true, abort these steps and ignore the newly created cookie entirely.
>
>   c. Update the creation-time of the newly-created cookie to match the creation-time of the old-cookie.
>
>   d. Remove the old-cookie from the cookie store.

2.5. Attaching a Partitioned Cookie to a Request

The following could be added to the first step of the algorithm in section 5.6.3 (Retrieval Algorithm):

> *If the cookie's partition-key is null, skip this step.

> Otherwise only include the cookie if the cookie's partition-key is same-site with the retrieval's cookie-partition-key.

3. Security Considerations

3.1. Partitioned requires Secure

This proposal takes the opportunity of defining the semantics of a new cookie attribute in order to require the Secure attribute, restricting this feature to secure protocols.

3.2. Partitioned cookies and XSS attacks

Sites are more susceptible to XSS attacks as embedded frames since these contexts rely on cross-site cookies for a notion of user session/state. Partitioning cross-site cookies makes them less vulnerable to being leaked via XSS, since an attacker would need to navigate the user's browser to the top-level site the cookie was created on in order for the browser to send the cookie at all.

3.3. Partitioned cookies and CSRF attacks

Cross-site cookies with the Partitioned attribute are less susceptible to CSRF attacks than unpartitioned, third-party cookies. This is because a partitioned cookie is only sent in requests when the browser is visiting the top-level site the cookie was created in, so a malicious top-level site will not be able to forge a request with an existing partitioned cookie unless they have compromised the top-level site that the cookie was sent from.

3.4. State proliferation for denial of service

Partitioning cross-site cookies inevitably will lead to more state proliferation on user's machines, so there is a possible DoS risk from partitioning cross-site cookies where a malicious embedded site could set many cookies across different partitions to take up memory on clients' machines. To help mitigate this, we suggest limiting the number of cookies a domain can set on a particular top-level site in the section below.

3.5. Partitioned cookies improve user privacy

The proposal suggests an alternate design for cross-site cookies which does not introduce a vector for cross-site tracking. This is a step towards making a larger privacy improvement for the web: removing third-party cookies.

3.6. Avoiding cross-partition leaks

One important privacy consideration is that partitioned cookies must not be subject to the 180 global per-domain cookie limit, otherwise they risk introducing a side channel for cross-site tracking. Instead, partitioned cookie limits should be counted separately

across different top-level sites to not leak any information about a user's activity on each respective site.

Another privacy consideration is that when a site sends the Clear-Site-Data header that contains "cookies", the user agent should only clear partitioned cookies whose partition key is same-site with the current top-level site. This will prevent abuse of partitioned cookies and the Clear-Site-Data header to establish identifiers that persist across different top-level sites.

Another privacy consideration is that the privacy guarantees of partitioned cookies can be circumvented by browser extensions with host permissions. Extensions' background contexts can query and store cookies across partitions, meaning they could store a cross-site identifier across partitions. Unfortunately, this type of attack is unavoidable due to the nature of extensions. Even if we block partitioned cookies (or even all cookies) from extensions' background contexts, an extension could still use content scripts to write cross-site identifiers to the DOM which the site's own script could copy to the site's partitioned cookie jar.

Finally, sites should be able to set partitioned cookies with the same name, domain, and path in different partitions. Otherwise, the presence or absence of a cookie with a particular name/domain/path would allow sites to learn about that user's activity on different top-level sites that make subresource requests to the cookie's domain.

4. Implementation Considerations

4.1. Applying Limits to Partitioned Cookie Jars

The following could be added as a new subsection of section 6.1 (Limits):

> User agents SHOULD enforce a separate per-domain limit for partitioned cookies for a particular cookie-partition-key. This limit SHOULD be lower than the per-domain limit for cookies without the Partitioned attribute to prevent cookies set on different top-level sites from reaching implementation memory limits. Since memory is the main concern, in addition to limiting the number of cookies a domain may use per partition, a user agent MAY limit how many bytes a domain's cookies occupy on the user agent's device to only 10 kilobytes per top-level partition. The user agent SHOULD consider memory occupied by each cookie to be the sum of the number of octets in the cookie-name and cookie-value.

4.2. Third-Party Cookie Controls

We may also want to add a paragraph about partitioned cookies to section 7.1 (Third-Party Cookies):

> Cross-site cookies which are set with the Partitioned attribute are only available on the top-level site in which they were created and therefore do not have the same privacy issues as other cross-site cookies. Due to this difference, user agents MAY exclude cross-site partitioned cookies from third-party cookie blocking controls.

4.3. Partitioned Cookies and Clear-Site-Data

We also can propose changes to the Clear-Site-Data header specification to prevent abuse of that header and partitioned cookies for cross-site tracking. The following could be added after step 2 in section 4.2.1 of [Clear-Site-Data]:

> 1. For each cookie in cookie-list, do the following:
>
>   a. If the cookie's cookie-partition-key attribute is null, skip this step.
>
>   b. Otherwise, if the top-document is not same-site with the cookie's partition-key then remove the cookie from cookie-list.

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[Clear-Site-Data] "Clear Site Data", <https://www.w3.org/TR/clear-site-data/>.

[HTML] WHATWG, "HTML", <https://html.spec.whatwg.org/>.

[RFC6265bis] Bingler, S., West, M., and J. Wilander, "Cookies: HTTP State Management Mechanism", Work in Progress, Internet-Draft, draft-ietf-httpbis-rfc6265bis-11, 7 November 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-rfc6265bis-11>.

6.2. Informative References

[CHIPS-Explainer] "Cookies Having Independent Partitioned State (CHIPS) explainer", <https://github.com/privacycg/CHIPS>.

Acknowledgments

The editors would also like to thank the following individuals for feedback, insight, and implementation of this draft and its predecessors (in alphabetical order): Kaustubha Govind, Johann Hofmann, Jeffrey Yasskin,

Author's Address

Dylan Cutler
Google LLC

Email: dylancutler@google.com