HTTPbis
Internet-Draft
Updates: 6265 (if approved)
Intended status: Standards Track
Expires: August 24, 2015

M. West
Google, Inc

February 20, 2015

First-Party-Only Cookies
draft-west-first-party-cookies-01

Abstract

This document updates RFC6265 by defining a "First-Party-Only" attribute which allows servers to assert that a cookie ought to be sent only in a "first-party" context. This assertion allows user agents to mitigate the risk of cross-site request forgery attacks, and other related paths to cross-origin information leakage.

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 http://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 August 24, 2015.

Copyright Notice

Table of Contents

1. Introduction

Section 8.2 of [RFC6265] eloquently notes that cookies are a form of ambient authority, attached by default to requests the user agent sends on a user's behalf. Even when an attacker doesn't know the contents of a user's cookies, she can still execute commands on the user's behalf (and with the user's authority) by asking the user agent to send HTTP requests to unwary servers.

Here, we update [RFC6265] with a simple mitigation strategy that allows servers to declare certain cookies as "First-party-only", meaning they should be attached to requests if and only if those requests occur in a first-party context. We define "first-party context" in terms of a user agent's top-level browsing context, which is the only security context a user can reasonably be expected to understand.

Note that the mechanism outlined here is backwards compatible with the existing cookie syntax. Servers may serve first-party cookies to all user agents; those that do not support the "First-Party-Only"

attribute will simply store a cookie which is returned in all applicable contexts, just as they do today.

1.1. Examples

First-party-only cookies are set via the "First-Party-Only" attribute in the "Set-Cookie" header field. That is, given a server's response to a user agent which contains the following header field:

Set-Cookie: SID=31d4d96e407aad42; First-Party-Only

Subsequent requests from that user agent can be expected to contain the following header field if and only if both the requested resource and the resource in the top-level browsing context match the cookie.

2. Terminology and notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234].

Two sequences of octets are said to case-insensitively match each other if and only if they are equivalent under the "i;ascii-casemap" collation defined in [RFC4790].

The terms "active document", and "top-level browsing context" are defined in the HTML Living Standard. [HTML]

The term "origin" and the mechanism of deriving an origin from a URI are defined in [RFC6454].

2.1. First-party and Third-party Requests

The URL displayed in a user agent's address bar is the only security context directly exposed to users, and therefore the only signal users can reasonably rely upon to determine whether or not they trust a particular website.

With that in mind, we define a "first-party" request as an HTTP request for a resource whose URL's origin matches the origin of the URL the user sees in the address bar. A "third-party" request is an HTTP request for a resource at any other origin.

To be more precise, given an HTTP request "request":

1. Let "context" be the top-level browsing context in the window responsible for "request".

2. Let "top-origin" be the origin of the location of the active document in "context".

3. If the origin of "request"'s URL is the same as "top-origin", "request" is a *first-party request*. Otherwise, "request" is a *third-party request*.

Note that we deal with the document's location in step 2 above, not with the document's origin. For example, a top-level document from "https://example.com" which has been sandboxed into a unique origin still creates a non-unique first-party context for subsequent requests.

This definition has a few implications:

- New windows create new first-party contexts.

- Full-page navigations create new first-party contexts. Notably, this includes both HTTP and "<meta>"-driven redirects.

- "<iframe>"s do not create new first-party contexts; their requests MUST be considered in the context of the origin of the URL the user actually sees in the user agent's address bar.

3. Server Requirements

This section describes extensions to [RFC6265] necessary to implement the server-side requirements of the "First-Party-Only" attribute.

3.1. Grammar

Add "First-Party-Only" to the list of accepted attributes in the "Set-Cookie" header field's value by replacing the "cookie-av" token definition in Section 4.1.1 of [RFC6265] with the following ABNF grammar:

```
cookie-av           = expires-av / max-age-av / domain-av /
                      path-av / secure-av / httponly-av /
                      first-party-only-av / extension-av
first-party-only-av = "First-Party-Only"
```

3.2. Semantics of the "First-Party-Only" Attribute (Non-Normative)

The "First-Party-Only" attribute limits the scope of the cookie such that it will only be attached to requests if those requests are "first-party", as described in Section 2.1. For example, requests for "https://example.com/sekrit-image" will attach first-party-only cookies if and only if the top-level browsing context is currently displaying a document from "https://example.com".

The changes to the "Cookie" header field suggested in Section 4.3 provide additional detail.

4. User Agent Requirements

This section describes extensions to [RFC6265] necessary in order to implement the client-side requirements of the "First-Party-Only" attribute.

4.1. The "First-Party" attribute

The following attribute definition should be considered part of the the "Set-Cookie" algorithm as described in Section 5.2 of [RFC6265]:

If the attribute-name case-insensitively matches the string "First-Party-Only", the user agent MUST append an attribute to the "cookie-attribute-list" with an "attribute-name" of "First-Party-Only" and an empty "attribute-value".

4.2. Monkey-patching the Storage Model

Note: There's got to be a better way to specify this. Until I figure out what that is, monkey-patching!

Alter Section 5.3 of [RFC6265] as follows:

1. Add "first-party-only-flag" to the list of fields stored for each cookie.

2. Before step 11 of the current algorithm, add the following:

    1. If the "cookie-attribute-list" contains an attribute with an "attribute-name" of "First-Party-Only", set the cookie's "first-party-only-flag" to true. Otherwise, set the cookie's "first-party-only-flag" to false.

    2. If the cookie's "first-party-only-flag" is set to true, and the request which generated the cookie is not a first-party

request (as defined in Section 2.1), then abort these steps and ignore the newly created cookie entirely.

4.3. Monkey-patching the "Cookie" header

Note: There's got to be a better way to specify this. Until I figure out what that is, monkey-patching!

Alter Section 5.4 of [RFC6265] as follows:

1. Add the following requirement to the list in step 1:

    * If the cookie's "first-party-only-flag" is true, then exclude the cookie if the HTTP request is a third-party request (see Section 2.1).

Note that the modifications suggested here concern themselves only with the origin of the top-level browsing context and the origin of the resource being requested. The cookie's "domain", "path", and "secure" attributes do not come into play for this comparison.

5. Authoring Considerations

5.1. Mashups and Widgets

The "First-Party-Only" attribute is inappropriate for some important use-cases. In particular, note that content intended for embedding in a third-party context (social networking widgets or commenting services, for instance) will not have access to first-party-only cookies. Non-first-party cookies may be required in order to provide seamless functionality that relies on a user's state.

Likewise, some forms of Single-Sign On might require authentication in a third-party context; these mechanisms will not function as intended with first-party-only cookies.

6. Privacy Considerations

First-party-only cookies in and of themselves don't do anything to address the general privacy concerns outlined in Section 7.1 of [RFC6265]. The attribute is set by the server, and serves to mitigate the risk of certain kinds of attacks that the server is worried about. The user is not involved in this decision. Moreover, a number of side-channels exist which could allow a server to link distinct requests even in the absence of cookies. Connection and/or socket pooling, Token Binding, and Channel ID all offer explicit methods of identification that servers could take advantage of.

We recommend, therefore, that servers interested in reducing the ambient authority of requests generated in a third-party context use such identification mechanisms only in addition to first-party-only cookies, and not as a replacement for them.

7. Security Considerations

7.1. Limitations

It is possible to bypass the protection that first-party-only cookies offer against cross-site request forgery attacks by creating first-party contexts in which to execute the attack. Consider, for instance, the URL "https://example.com/logout" which logs the current user out of "example.com". If the user's session cookie is a first-party-only cookie, then embedding the logout URL in an "<iframe>" element or an "<img>" element won't log her out, as the cookie won't be sent. Popping up a new window, or triggering a top-level navigation, on the other hand, will create a first-party context, attach cookies, and perform the logout.

Note, though, that popping up a window, or doing a top-level navigation are both significantly more visible to the user than loading a subresource. Users will at least have the opportunity to notice that something strange is going on, which hopefully reduces an attacker's ability to perform untargeted attacks.

Further, note that certain kinds of attacks are no longer possible if a first-party context is required. Information leakage attacks which rely on visible side-effects of loading a session-protected image, for example, can no longer access those side-effects if the image is loaded in a new window. Timing attacks like those Paul Stone outlines in [pixel-perfect] are no longer possible if the session cookie is first-party-only, as they rely on "<iframes>" to contain the protected content in a way the attacker can manipulate.

8. Acknowledgements

The first-party cookie concept documented here is indebited to Mark Goodwin's and Joe Walker's [samedomain-cookies].

9. References

9.1. Normative References

[HTML] Hickson, I., "HTML Living Standard", n.d., <https://html.spec.whatwg.org/>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4790] Newman, C., Duerst, M., and A. Gulbrandsen, "Internet Application Protocol Collation Registry", RFC 4790, March 2007.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, December 2011.

9.2. Informative References

[pixel-perfect] Stone, P., "Pixel Perfect Timing Attacks with HTML5", n.d., <http://www.contextis.com/documents/2/Browser_Timing_Attacks.pdf>.

[samedomain-cookies] Goodwin,, M. and J. Walker, "SameDomain Cookie Flag", 2011, <http://people.mozilla.org/~mgoodwin/SameDomain/samedomain-latest.txt>.

Author's Address

Mike West
Google, Inc

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