HTTPbis
Internet-Draft
Updates: 6265 (if approved)
Intended status: Standards Track
Expires: October 22, 2015

M. West
Google, Inc

April 20, 2015

Origin Cookies
draft-west-origin-cookies-01

Abstract

This document updates RFC6265, defining the "origin" attribute for cookies and the "Origin-Cookie" header field, which together allow servers to choose to harmonize the security policy of their cookies with the same-origin policy which governs other available client-side storage mechanisms.

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 October 22, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Cookies, as defined by [RFC6265], diverge from the web's general security policy in a number of ways which may be surprising to implementers and authors who haven't carefully read that document's discussion of "domain matching", and "path matching", or who ignored the admonitions regarding "Weak Confidentiality" and "Weak Integrity".

This document updates [RFC6265], describing a mechanism by which servers can opt-in to harmonizing cookies' security policy with the same-origin policy, as described in [RFC6454]. User agents that support these "origin cookies" will ignore a "Set-Cookie" header's value's "Path", "Domain", and "Secure" attributes if an "Origin" attribute is present, instead tying the cookie to the origin that set it. These "origin cookies" will be returned in a new "Origin-Cookie" header field (see Section 4.4 for detail), separating them from non-origin cookies in a way a server can easily distinguish.

Harmonizing with the same-origin policy mitigates the confidentiality and integrity risks noted above by ensuring that origin cookies are not influenced by malicious code running on a server's subdomain or a non-standard port or scheme.

Note that the mechanism outlined here is backwards compatible with the existing cookie syntax. Servers may serve origin cookies to all user agents; those that do not support the "Origin" attribute will simply store a non-origin cookie, just as they do today.

1.1. Examples

Origin cookies are set via the "Origin" 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; Secure; HttpOnly; Origin
```

Subsequent requests from that user agent can be expected to contain the following header field:

```
Origin-Cookie: SID=31d4d96e407aad42
```

Non-origin cookies are returned in the "Cookie" header field as usual. If both non-origin and origin cookies are present for an origin, then both a "Cookie" and "Origin-Cookie" header field will be present. That is, given a server's response to a user agent which contains the following header fields:

```
Set-Cookie: SID=31d4d96e407aad42; Origin
Set-Cookie: lang=en-US;
```

Subsequent requests from that user agent can be expected to contain the following header fields:

```
Cookie: lang=en-US
Origin-Cookie: SID=31d4d96e407aad42
```

User agents that support origin cookies are required to advertise their support for such by sending an "Origin-Cookie" header whenever a "Cookie" header is sent. That is, given the following server response:

```
Set-Cookie: lang=en-US; Secure; HttpOnly
```

Subsequent requests from a user agent that supports origin cookies can be expected to contain the following header fields:

```
Cookie: lang=en-US
Origin-Cookie:
```

Note that the "Origin-Cookie" field is empty.

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].

3. Server Requirements

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

3.1. Grammar

Add "Origin" 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 /
            origin-av / extension-av
origin-av = "origin"
```

3.2. Semantics of the "Origin" Attribute (Non-Normative)

The "Origin" attribute limits the scope of the cookie such that it will only be attached to requests if those request match the origin which set the cookie. For example, requests for "https://example.com/" will attach origin cookies if and only if those cookies were set by "https://example.com/".

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

3.3. The <spanx style="verb">Origin-Cookie</spanx> header

3.3.1. Syntax

The user agent sends stored origin cookies to the origin server in the "Origin-Cookie" header. If the server conforms to the requirements in Section 3 (and the user agent conforms to the requirements in Section 4), the user agent will send an "Origin-Cookie" header which conforms to the following grammar:

origin-cookie-header = "Origin-Cookie:" OWS [cookie-string] OWS

3.3.2. Semantics

The semantics of the "cookie-string" are the same as those of the same token in the "Cookie" header.

Note, however, that the "Origin-Cookie" header MAY be empty, and MUST be sent with every request, even if no origin cookies are present in the cookie store. This allows conformant servers to detect a user agent's support for origin cookies, and therefore to make a secure decision about whether or not to fallback to searching through the "Cookie" header for specific cookies. See Section 5.2 for details.

4. User Agent Requirements

This section describes extensions to [RFC6265] necessary in order to implement the client-side requirements of the "Origin" attribute and "Origin-Cookie" header field.

4.1. The "Origin" 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 "Origin", the user agent MUST append an attribute to the cookie-attribute-list with an attribute-name of "Origin" 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 "origin" and "origin-flag" to the list of fields stored about 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 "Origin":

        1. Set the cookie's "domain" attribute to the empty string.

        2. Set the cookie's "host-only-flag" to true.

        3. Set the cookie's "origin" to the origin of "request-uri", as defined by Section 4 of [RFC6454].

        4. Set the cookie's "origin-flag" to true.

        5. Set the cookie's "path" attribute to the empty string.

        6. Set the cookie's "secure-only-flag" to false.

        Otherwise: set the cookie's "origin-flag" to false, and its "origin" to "null".

    2. If the newly created cookie's "origin-flag" is set to true, and the cookie store contains a cookie with the same"name", "origin", and "origin-flag" as the newly created cookie:

        1. Let "old-cookie" be the existing cookie with the same "name", "origin", and "origin-flag" as the newly created cookie.

        2. Update the "creation-time" of the newly created cookie to match the "creation-time" of "old-cookie".

        3. Remove "old-cookie" from the cookie store.

3. Change the priority order for excess cookie removal to the following:

    1. Expired cookies.

    2. Cookies whose "origin-flag" is false that share a "domain" field with more than a predetermined number of other cookies.

    3. Cookies whose "origin-flag" is true that share a "domain" field with more than a predetermined number of other cookies.

    4. Cookies whose "origin-flag" is false.

    5. All cookies.

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:

    * The cookie's "origin-flag" is false.

4.4. The "Origin-Cookie" header field

The user agent includes stored cookies whose "origin-flag" is set in the "Origin-Cookie" request header. When the user agent generates an HTTP request, it MUST NOT attach more than one "Origin-Cookie" header field.

A user agent MAY omit the "Origin-Cookie" header in its entirety. For example, the user agent may wish to block sending cookies during "third-party" requests. If, however, a "Cookie" header is sent, a user agent MUST send an "Origin-Cookie" header.

If the user agent does attach an "Origin-Cookie" header field to an HTTP request, the user agent MUST send the "cookie-string" as defined below as the value of the header field.

The user agent MUST use an algorithm equivalent to the following algorithm to compute the "cookie-string" from a cookie store and a "request-uri":

1. Let "cookie-list" be the set of cookies from the cookie store that meets all the following requirements:

    * The cookie's "origin-flag" is true.

    * The cookie's "origin" matches the origin of "request-uri". [RFC6454]

2. The user agent SHOULD sort the "cookie-list" in the following order:

    * Cookies with earlier "creation-time"s are listed before cookies with later "creation-time"s.

3. Update the "last-access-time" of each cookie in the "cookie-list" to the current date and time.

4. Serialize the "cookie-list" into a "cookie-string" by processing each cookie in the "cookie-list" in order:

    1. Output the cookie's "name", the %x3D ("=") character, and the cookie's "value".

    2. If there is an unprocessed cookie in the "cookie-list", output the characters %x3B and %x20 ("; ").

5. Security Considerations

The security considerations listed in Section 8 of [RFC6265] apply equally to origin cookies, with the exceptions of Sections 8.6 ("Weak Confidentiality") and Sections 8.7 ("Weak Isolation"), both of which are substantially improved if the "Origin" attribute is set. Further:

5.1. Paths are ignored

Origin cookies will break the (flawed) "Path"-based isolation strategy which some servers may be attempting to implement. If a server has used the "Path" attribute to limit cookies to specific areas of a site (say "/admin"), then they may be surprised by origin cookies' pathless behavior.

That said, paths offer little to no protection against malicious code. The origin is the only security boundry enforced rigorously by user agents; it is therefore the only security boundry that server operators ought to rely on for isolation.

5.2. Downgrade attacks

If a server chooses to scan both the "Origin-Cookie" and "Cookie" headers in order to provide backwards compatibility with user agents that don't support origin cookies, it ought to be done carefully. Careless fallback strategies can provide a window of opportunity for an attacker to inject cookies with the same name as origin cookies from a subdomain, bypassing origin cookies' main advantage.

- If the "Origin-Cookie" header is present, servers SHOULD NOT check the "Cookie" header for cookies which it set as origin cookies.

- If a user agent is known to support origin cookies, servers SHOULD check only the "Origin-Cookie" header for origin cookies, and SHOULD NOT fallback to the "Cookie" header if the "Origin-Cookie" header is not present, or if a particular cookie is not found there.

6. IANA Considerations

The permanent message header field registry (see [RFC3864]) shall be updated with the following registration:

6.1. Origin-Cookie

- Header field name: Origin-Cookie
- Applicable protocol: http
- Status: standard
- Author/Change controller: IETF
- Specification document: This specification (see Section 4.4)

7. References

7.1. Normative References

[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.

7.2. Informative References

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[draft-abarth-cake-01]
Barth, A., "Origin Cookies", September 2011, <https://tools.ietf.org/html/draft-abarth-cake-01>.

[origin-cookies-w2sp]
Bortz,, A., Barth, A., and A. Czeskis, "Origin Cookies: Session Integrity for Web Applications", 2011, <http://w2spconf.com/2011/papers/session-integrity.pdf>.

Appendix A. Acknowledgements

The origin cookie concept documented here is heavily indebted to and based upon Adam Barth's [draft-abarth-cake-01] document, as well as Andrew Bortz, Adam Barth, and Alexei Czeskis' paper [origin-cookies-w2sp].

Author's Address

Mike West
Google, Inc

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