HTTPbis
Internet-Draft
Updates: 6265 (if approved)
Intended status: Standards Track
Expires: July 10, 2016

M. West
Google, Inc
January 7, 2016

Deprecate modification of 'secure' cookies from non-secure origins
draft-west-leave-secure-cookies-alone-05

Abstract

This document updates RFC6265 by removing the ability for a non-secure origin to set cookies with a 'secure' flag, and to overwrite cookies whose 'secure' flag is set. This deprecation improves the isolation between HTTP and HTTPS origins, and reduces the risk of malicious interference.

Status of This Memo

This Internet-Draft will expire on July 10, 2016.

Copyright Notice

Table of Contents

1. Introduction

Section 8.5 and Section 8.6 of [RFC6265] spell out some of the drawbacks of cookies' implementation: due to historical accident, non-secure origins can set cookies which will be delivered to secure origins in a manner indistinguishable from cookies set by that origin itself. This enables a number of attacks, which have been recently spelled out in some detail in [COOKIE-INTEGRITY].

We can mitigate the risk of these attacks by making it more difficult for non-secure origins to influence the state of secure origins. Accordingly, this document recommends the deprecation and removal of non-secure origins' ability to write cookies with a 'secure' flag, and their ability to overwrite cookies whose 'secure' flag is set.

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].

The "scheme" component of a URI is defined in Section 3 of [RFC3986].

3. Recommendations

This document updates Section 5.3 of [RFC6265] as follows:

1. After step 8 of the current algorithm, which sets the cookie's "secure-only-flag", execute the following step:

    1. If the "scheme" component of the "request-uri" does not denote a "secure" protocol (as defined by the user agent),

and the cookie's "secure-only-flag" is "true", then abort these steps and ignore the newly created cookie entirely.

2. Before step 11, execute the following step:

    1. If the newly created cookie's "secure-only-flag" is not set, and the "scheme" component of the "request-uri" does not denote a "secure" protocol, then abort these steps and ignore the newly created cookie entirely if the cookie store contains one or more cookies that meet all of the following criteria:

        1. Their "name" matches the "name" of the newly created cookie.

        2. Their "secure-only-flag" is set.

        3. Their "domain" domain-matches the "domain" of the newly created cookie, or vice-versa.

        Note: This comparison intentionally ignores the "path" component. The intent is to allow the "secure" flag to supercede the "path" restrictions to protect sites against cookie fixing attacks.

        Note: This allows "secure" pages to override "secure" cookies with non-secure variants. Perhaps we should restrict that as well?

3. In order to ensure that a non-secure site can never cause a "secure" cookie to be evisted, adjust the "remove excess cookies" priority order at the bottom of Section 5.3 to be the following:

    1. Expired cookies.

    2. Cookies whose "secure-only-flag" is not set and which share a "domain" field with more than a predetermined number of other cookies.

    3. Cookies that share a "domain" field with more than a predetermined number of other cookies.

    4. All cookies.

    Note that the eviction algorithm specified here is triggered only after insertion of a cookie which causes the user agent to exceed some predetermined upper bound. Conforming user agents MUST

ensure that inserting a non-secure cookie does not cause a secure cookie to be removed.

4. Security Considerations

This specification increases a site's confidence that secure cookies it sets will remain unmodified by insecure pages on hosts which it domain-matches. Ideally, sites would use HSTS as described in [RFC6797] to defend more robustly against the dangers of non-secure transport in general, but until adoption of that protection becomes ubiquitous, this deprecation this document recommends will mitigate a number of risks.

The mitigations in this document do not, however, give complete confidence that a given cookie was set securely. If an attacker is able to impersonate a response from "http://example.com/" before a user visits "https://example.com/", the user agent will accept any cookie that the insecure origin sets, as the "secure" cookie won't yet be present in the user agent's cookie store. An active network attacker may still be able to use this ability to mount an attack against "example.com", even if that site uses HTTPS exclusively.

The proposal in [COOKIE-PREFIXES] could mitigate this risk, as could "preloading" HSTS for "example.com" into the user agent [HSTS-PRELOADING].

5. References

5.1. Normative References

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

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

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

5.2. Informative References

[COOKIE-INTEGRITY]
Zheng, X., Jiang, J., Liang, J., Duan, H., Chen, S., Wan, T., and N. Weaver, "Cookies Lack Integrity: Real-World Implications", n.d., <https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-zheng.pdf>.

[COOKIE-PREFIXES]
West, M., "Cookie Prefixes", n.d., <https://tools.ietf.org/html/draft-west-cookie-prefixes>.

[HSTS-PRELOADING]
"HSTS Preload Submission", n.d., <https://hstspreload.appspot.com/>.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <http://www.rfc-editor.org/info/rfc6797>.

Appendix A. Acknowledgements

Richard Barnes encouraged a formalization of the deprecation proposal. [COOKIE-INTEGRITY] was a useful exploration of the issues [RFC6265] described.

Author's Address

Mike West
Google, Inc

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