HTTPbis
Internet-Draft
Updates: 6265 (if approved)
Intended status: Standards Track
Expires: April 14, 2016
October 12, 2015

M. West
Google, Inc

Cookie Prefixes
draft-west-cookie-prefixes-04

Abstract

This document updates RFC6265 by adding a set of restrictions upon the names which may be used for cookies with specific properties. These restrictions enable user agents to smuggle cookie state to the server within the confines of the existing "Cookie" request header syntax, and limits the ways in which cookies may be abused in a conforming user agent.

Status of This Memo

This Internet-Draft will expire on April 14, 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, it is impossible for a server to have confidence that a cookie set in a secure way (e.g., as a domain cookie with the "Secure" (and possibly "HttpOnly") flags set) remains intact and untouched by non-secure subdomains.

We can't alter the syntax of the "Cookie" request header, as that would likely break a number of implementations. This rules out sending a cookie's flags along with the cookie directly, but we can smuggle information along with the cookie if we reserve certain name prefixes for cookies with certain properties.

This document describes such a scheme, which enables servers to set cookies which conforming user agents will ensure are "Secure", and locked to a domain.

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. Prefixes

3.1. The "$Secure-" prefix

If a cookie's name begins with "$Secure-", the cookie MUST be:

1. Set with a "Secure" attribute

2. Set from a URI whose "scheme" is considered "secure" by the user agent.

The following cookie would be rejected when set from any origin, as the "Secure" flag is not set

```
Set-Cookie: $Secure-SID=12345; Domain=example.com
```

While the following would be accepted if set from a secure origin (e.g. "https://example.com/"), and rejected otherwise:

```
Set-Cookie: $Secure-SID=12345; Secure; Domain=example.com
```

3.2. The "$Host-" prefix

If a cookie's name begins with "$Host-", the cookie MUST be:

1. Set with a "Secure" attribute

2. Set from a URI whose "scheme" is considered "secure" by the user agent.

3. Sent only to the host which set the cookie. That is, a cookie named "$Host-cookie1" set from "https://example.com" MUST NOT contain a "Domain" attribute (and will therefore be sent only to "example.com", and not to "subdomain.example.com").

4. Sent to every request for a host. That is, a cookie named "$Host-cookie1" MUST contain a "Path" attribute with a value of "/".

The following cookies would always be rejected:

```
Set-Cookie: $Host-SID=12345
Set-Cookie: $Host-SID=12345; Secure
Set-Cookie: $Host-SID=12345; Domain=example.com
Set-Cookie: $Host-SID=12345; Domain=example.com; Path=/
Set-Cookie: $Host-SID=12345; Secure; Domain=example.com; Path=/
```

While the following would be accepted if set from a secure origin (e.g. "https://example.com/"), and rejected otherwise:

```
Set-Cookie: $Host-SID=12345; Secure; Path=/
```

4. User Agent Requirements

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

After step 10 of the current algorithm, the cookies flags are set. Insert the following steps to perform the prefix checks this document specifies:

1. If the "cookie-name" begins with the string "$Secure-" or "$Host-", abort these steps and ignore the cookie entirely unless both of the following conditions are true:

   * The cookie's "secure-only-flag" is "true"

   * "request-uri"'s "scheme" component denotes a "secure" protocol (as determined by the user agent)

2. If the "cookie-name" begins with the string "$Host-", abort these steps and ignore the cookie entirely unless the following conditions are true:

   * The cookie's "host-only-flag" is "true"

   * The cookie's "path" is "/"

5. Aesthetic Considerations

Prefixes are ugly. :(

6. Security Considerations

6.1. Secure Origins Only

It would certainly be possible to extend this scheme to non-secure origins (and an earlier draft of this document did exactly that). User agents, however, are slowly moving towards a world where features with security implications are available only over secure transport (see [SECURE-CONTEXTS], [POWERFUL-FEATURES], and [DEPRECATING-HTTP]). This document follows that trend, limiting exciting new cookie properties to secure transport in order to ensure that user agents can make claims which middlemen will have a hard time violating.

To that end, note that the requirements listed above mean that prefixed cookies will be rejected entirely if a non-secure origin attempts to set them.

6.2. Limitations

This scheme gives no assurance to the server that the restrictions on cookie names are enforced. Servers could certainly probe the user agent's functionality to determine support, or sniff based on the "User-Agent" request header, if such assurances were deemed necessary.

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.

[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, April 2011.

7.2. Informative References

[DEPRECATING-HTTP]
Barnes, R., "Deprecating Non-Secure HTTP", n.d., <https://blog.mozilla.org/security/2015/04/30/deprecating-non-secure-http/>.

[Lawrence2015]
Lawrence, E., "Duct Tape and Baling Wire -- Cookie Prefixes", n.d., <http://textslashplain.com/2015/10/09/duct-tape-and-baling-wirecookie-prefixes/>.

[POWERFUL-FEATURES]
Palmer, C., "Prefer Secure Origins for Powerful New Features", n.d., <https://www.chromium.org/Home/chromium-security/prefer-secure-origins-for-powerful-new-features>.

[SECURE-CONTEXTS]
West, M., "Secure Contexts", n.d., <https://w3c.github.io/webappsec-secure-contexts/>.

Appendix A. Acknowledgements

Eric Lawrence had this idea a million years ago, and wrote about its genesis in [Lawrence2015]. Devdatta Akhawe helped justify the potential impact of the scheme on real-world websites.

Author's Address

Mike West
Google, Inc

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