Network Working Group
Internet-Draft
Updates: 6454 (if approved)
Intended status: Standards Track
Expires: August 5, 2012

Y. Nir, Ed.
Check Point

February 2, 2012

A More Granular Web Origin Concept
draft-nir-websec-extended-origin-00

Abstract

This document defines an HTTP header that allows to partition a single origin as defined in RFC 6454 into multiple origins, so that the same origin policy applies among them.

The header introduced in this document allows the portal to specify that resources that appear to be from the same origin should, in fact, be treated as though they are from different origins, by extending the 3-tuple of the origin to a 4-tuple. The user agent is expected to apply the same-origin policy according to the 4-tuple rather than the 3-tuple.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Web portals such as SSL VPNs "flatten" the Web by providing access to multiple web sites through a single host. For example, a company portal may be located at https://sslvpn.example.com, and allow remote access to several websites that form the corporate intranet as well as webified access to the mail server. The different services are distinguised by implementation-specific manipulation of the URL. For example, the following three URLs may be respectively for the internal mail server, for the internal wiki, and for Wikipedia:

1. https://sslvpn.example.com/link/my_web_mail/inbox/index.html
2. https://sslvpn.example.com/link/the_wiki/index.html
3. https://sslvpn.example.com/ext/wikipedia.org

The problem here is that although there are separate servers, they all map to the same origin as defined in [RFC6454]. Scripts from any of these sites can affect others. In fact, the Origin header as defined in section 7 of RFC 6454 can leak information to the real web server that it is located within the same flattened domain.

The HTTP header introduced in this document allows the portal to specify that URLs that appear to be from the same origin are, in fact, from different origins, by extending the 3-tuple of the origin to a 4-tuple. The user agent would be expected to apply the same-origin policy according to the 4-tuple rather than the 3-tuple.

1.1. Conventions Used in This Document

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].

2. The Extended-Origin Header

When a web portal hides multiple actual web sites behind its own origin, it MUST add the new Extended-Origin header defined in the next section. The name field need not be related to the actual web origin, and is not meant for human consumption. The requirement is only that different origins MUST have different names in the header.

If the response from the original web site already contains one or more Extended-Origin headers, then the portal adds its own header after the rest.

2.1. Header Format

The ABNF is to be added.

The header includes a name, which is not necessarily meant for human consumption, and an optional path parameter. The general format is

```
Extended-Origin: name[; path=/something]
```

This means that all requests of the format "GET /something/..." will be considered as going to the origin defined by the combination of the RFC 6454 origin and the name. As such, cookies from the portal MUST not be returned in requests to the extended origin, and vice versa. Scripts from inside the extended origin MUST be prevented from executing requests against the main portal and against other extended origins within the same portal.

2.2. Update to the Serialization Requirements

Section 6 of RFC 6454 defines how to serialize an origin for inclusion in the "Origin" header defined in section 7 of that RFC.

For serializing an extended origin, follow steps 1-5 of section 6.1 or 6.2 of RFC 6454. To the result, append a U+0023 code point (number sign - #) and the content of the Extended-Origin header. Return the result

If the response contains more than one Extended-Origin header, then the user agent MUST append the content of all, separated by number symbols, in reverse order. For example, if the server response looks like this:

```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
Extended-Origin: webmail
Extended-Origin: some_other_portal
```

Then the origin should be as follows:

```
https://sslvpn.example.com#some_other_portal#webmail
```

3. Examples

Here's an example of a connection with both the Extended-Origin and the Origin headers.

```
CONNECT  https://sslvpn.example.com

GET / HTTP/1.1

HTTP/1.1 200 OK
Content-Type: application/octet-stream
Set-Cookie: session=1234

<html>
  <body>
    Welcome, you can read your mail
      <a href="/link/my_web_mail/inbox/index.html">here</a>
  </body>
</html>

GET /link/my_web_mail/inbox/index.html HTTP/1.1
Referer: https://sslvpn.example.com/
Cookie: session=1234

HTTP/1.1 200 OK
Content-Type: application/octet-stream
Extended-Origin: my_web_mail; path=/link/my_web_mail
Set-Cookie: mailsession=5678

<html>
  <body>
    You have unread message. Jumping there in 5 seconds.
    <script>...</script>
  </body>
</html>

GET /link/my_web_mail/inbox/msg0945.html HTTP/1.1
Referer: https://sslvpn.example.com/link/my_web_mail/inbox/index.htm
Origin: https://sslvpn.example.com#my_web_mail
Cookie: mailsession=5678
```

In this example, the first GET was the result of the user typing in an address, or following a link. Therefore it has no Origin header. It goes to the main page of the portal, so the response contains no Extended-Origin.

The second GET also happened because of clicking a link, not by any action of the page, so there's no need to send an Origin header. If there had been such a header, it would be just as defined in RFC 6454: https://sslvpn.example.com

The third GET is caused by a script running on the mail page. This

page came with an Extended-Origin header, and so the user agent constructs the Origin header in the request according to the new rules in Section 2.2.

Note that the cookie set by the main portal was not sent in the third request, because it the second reply belongs to a different origin, and the request URL matches the path parameter of the Extended-Origin header.

A more complex example is when the portal hides another portal, resulting in two Extended-Origin headers. Shown here:

```
CONNECT  https://sslvpn.example.com

GET /link/someotherportal/mail/index.html HTTP/1.1
Referer: https://sslvpn.example.com/mainpage.html
Origin: https://sslvpn.example.com

HTTP/1.1 200 OK
Content-Type: application/octet-stream
Extended-Origin: webmail; path=/link/someotherportal/mail
Extended-Origin: some_other_portal; path=/link/webmail
Set-Cookie: session=90ab

<html>
  <body>
    You have unread message. Jumping there in 5 seconds.
    <script>...</script>
  </body>
</html>


GET /link/someotherportal/my_web_mail/inbox/msg0945.html HTTP/1.1
Origin: https://sslvpn.example.com#some_other_portal#webmail
Cookie: session-90ab
```

In this example we see that only the first path parameter is considered. The cookies are sent whenever the link matches the first path parameter.

4. CORS interaction

The interaction between this draft and CORS ([CORS]) is to be added.

5. Open Issues

5.1. Other Methods of Encoding Server Identity

Some SSL-VPN products and configurations do not encode the server identity using a prefix in the URL, as shown in the example in Section 3. One such Method is this:

```
 https://sslvpn.example.com/p/inb/msg0945.html,HOST=mail.example.com
```

The issue here is that the way the path parameter is defined, you cannot use it to define what URLs belong to the extended origin. We could replace it with a parameter that accepts a regular expression, but that seems overly complex:

```
    Extended-Origin: webmail; expr=/p/*,HOST=mail.example.com
```

6. Acknowledgements

Oren Souroujon contributed some of the text in this document, and also came up with the original idea. Yehezkel Horowitz helped with reviewing the draft and pointing out the issues with cookies and paths.

7. Security Considerations

This document causes compliant clients to disallow certain actions that are allowed today. In that sense, it reduces the attack surface.

More to be added.

8. IANA Considerations

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

- o Header field name: Extended-Origin
- o Applicable protocol: http
- o Status: Standard
- o Author/Change controller: IETF
- o Specification document: this specification

9. Changes from Previous Versions

First version

10. References

10.1. Normative References

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

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

10.2. Informative References

[CORS] van Kesteren, A., "Cross-Origin Resource Sharing", W3C Working Draft WD-cors-20100727, July 2010.

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

Author's Address

Yoav Nir (editor)
Check Point Software Technologies Ltd.
5 Hasolelim st.
Tel Aviv 67897
Israel

Email: ynir@checkpoint.com