

Network Working Group
Internet-Draft
Updates: [6454](#) (if approved)
Intended status: Standards Track
Expires: September 7, 2012

Y. Nir, Ed.
Check Point
March 6, 2012

A More Granular Web Origin Concept
draft-nir-websec-extended-origin-02

Abstract

This document defines an HTTP header that allows the partitioning of 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 a 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. A compliant user agent is expected to apply the same-origin policy according to the 4-tuple rather than the 3-tuple.

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 September 7, 2012.

Copyright Notice

Copyright (c) 2012 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

Internet-Draft

Extended Origin

March 2012

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	3
1.1.	Conventions Used in This Document	3
2.	The Extended-Origin Header	3
2.1.	Header Format	4
2.2.	Update to the Serialization Requirements	4
3.	Examples	4
4.	Determining the Extended Origin based on a URL	6
5.	CORS interaction	7
6.	Open Issues	7
6.1.	Other Methods of Encoding Server Identity	7
7.	Acknowledgements	8
8.	Security Considerations	8
9.	IANA Considerations	8
10.	Changes from Previous Versions	8
10.1.	Changes in version -02	8
10.2.	Changes in version -01	8
11.	References	9
11.1.	Normative References	9
11.2.	Informative References	9
	Author's Address	9

Internet-Draft

Extended Origin

March 2012

[1.](#) Introduction

Reverse proxies such as SSL VPN portals "flatten" part of 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 distinguished 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 a 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 1-3 of [section 6.1](#) or 6.2 of [RFC 6454](#). To the result, append the name from the Extended-Origin header and a U+002E FULL STOP code points ("."). Then continue with steps 4-6.

For example, if the host is sslvpn.example.com, and the name in the extended origin header is webmail, then the serialized origin becomes

<https://webmail.sslvpn.example.com>

To avoid collisions between serialized extended origins and serialized non-extended origins, servers SHOULD NOT use readable origins such as "webmail". Instead they should choose random-looking extended origin names, possibly obtained by hashing an internally meaningful name.

3. Examples

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

Nir

Expires September 7, 2012

[Page 4]

Internet-Draft

Extended Origin

March 2012

```
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: d41d8cd98f00b204; path=/link/my_web_mail
```

Set-Cookie: mailsession=5678

```
<html>
  <body>
    You have 1 unread message. Jumping 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://d41d8cd98f00b204.sslvpn.example.com>
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. The second reply marked all requests beginning with "/link/my_web_mail" as belonging to the extended origin, and the third request matches that pattern. Cookies from the non-extended origin are not forwarded to the extended origin.

The second request did include the portal cookie in a request to the mail server. This is only an issue with the main portal cookies, not among the extended origins. Some SSL VPN portals strip their own cookies from requests going to the other servers, and this behavior is RECOMMENDED.

[4.](#) Determining the Extended Origin based on a URL

This section defines an algorithm for converting a URL into an origin. This section is not normative, and compliant browsers may implement this in other ways.

For each visited site, the browser keeps a table mapping paths to origin names. Initially, this table looks like this:

Path	Name
/	

Table 1: Initial table

As Extended-Origin headers are encountered, entries are added to the table. For example, after seeing the header in the example in [Section 3](#), the table will look like this:

Path	Name
/	
/link/my_web_mail	d41d8cd98f00b204
/link/SAP	12c30f3bb3275376

Table 2: The table with 2 more entries

When presented with a URL, the browser can normally figure the scheme, host and port. The name parameter can be figured out from the path according to the closes match in the table. Here are some URLs and the origins to which they map:

URL	Extended Origin
-----	-----------------

https://sslvpn.example.com/index.html	https://sslvpn.example.com
https://sslvpn.example.com/link/my_web_mail/msg0005.html	https://d41d8cd98f00b204.sslvpn.example.com
https://sslvpn.example.com/link/SAPIENCE/index.html	https://sslvpn.example.com
https://sslvpn.example.com/link/SAP/index.html	https://12c30f3bb3275376.sslvpn.checkpoint.com
https://sslvpn.example.com/ext/wikipedia.org/index.html	https://sslvpn.example.com

Table 3: Extended origin examples

5. CORS interaction

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

6. Open Issues

6.1. 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

7. 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.

Thanks to James Manger and Tobias Gondrom for reviewing the first version of this draft.

8. 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.

9. 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

10. Changes from Previous Versions

NOTE TO RFC EDITOR: Please remove this section before publication.

10.1. Changes in version -02

Added [Section 4](#) about converting URLs to extended origins

10.2. Changes in version -01

Removed the special handling of portals behind portals.

Changed the syntax of the serialized origin from fragment-like to subdomain-like.

Cleaned up some grammar.

[11.](#) References

[11.1.](#) 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.

[11.2.](#) 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

Nir

Expires September 7, 2012

[Page 9]