WEBSEC
Internet-Draft
Intended status: Informational
Expires: February 28, 2014

D. Ross
Microsoft
T. Gondrom
Thames Stanley

August 27, 2013

HTTP Header Field X-Frame-Options
draft-ietf-websec-x-frame-options-12

Abstract

To improve the protection of web applications against Clickjacking, this definition describes the X-Frame-Options HTTP response header field that declares a policy communicated from the server to the client browser on whether the browser may display the transmitted content in frames that are part of other web pages. This informational document serves to document the existing use and specification of this X-Frame-Options HTTP response header field.

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 February 28, 2014.

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. X-Frame-Options Header 3
2.1. Syntax 3
2.2. Augmented Backus-Naur Form (ABNF) 5
2.2.1. Examples of X-Frame-Options 5
2.3. Design Issues 5
2.3.1. Enable HTML content from other domains 6
2.3.2. Browser Behaviour and Processing 6
2.3.2.1. Violation of X-Frame-Options 6
2.3.2.2. Variation in current browser behaviour 6
2.3.2.3. Usage design pattern and example scenario for the ALLOW-FROM parameter 8
2.3.2.4. No caching of the X-Frame-Options header 8
3. Acknowledgements 9
4. IANA Considerations 9
4.1. Registration Template 9
5. Security Considerations 9
5.1. Privacy Considerations 10
6. References 10
6.1. Normative References 10
6.2. Informative References 11
Appendix A. Browsers that support X-Frame-Options 12
Appendix B. Description of a Clickjacking attack 12
B.1. Shop 12
B.2. Online Shop Confirm Purchase Page 12
B.3. Flash Configuration 13
Authors' Addresses 13

1. Introduction

In 2009 and 2010 many browser vendors ([Microsoft-X-Frame-Options], [CLICK-DEFENSE-BLOG], [Mozilla-X-Frame-Options]) introduced the use of a non-standard HTTP [RFC2616] header field "X-Frame-Options" to protect against Clickjacking [Clickjacking]. HTML-based web applications can embed or "frame" other web pages. Clickjacking is a type of attack that occurs when an attacker uses multiple transparent or opaque layers in the user interface to trick a user into clicking on a button or link on another page from server B when they were intending to click on the same place of the overlaying page from server A. Thus, the attacker is "hijacking" clicks meant for their page A and routing them to another page B. The attacker is tricking

the user (who sees the overlaying user interface content from page A) into clicking specific locations on the underlying page from server B, triggering some actions on server B and potentially using an existing session context in that step. This is an attack on both the user and on server B. And server A may or may not be the attacker.

This specification provides informational documentation about the current use and definition of the X-Frame-Options HTTP header field. As described in Section 2.3.2.2 not all browsers implement X-Frame-Options exactly in the sames way, which can lead to unintended results. And given that the "X-" construction is deprecated [RFC6648], the X-Frame-Options header field will in the future be replaced by the Frame-Options directive in the Content Security Policy Version 1.1 [CSP-1-1].

Existing anti-ClickJacking measures, e.g. Frame-breaking Javascript, have weaknesses so that their protection can be circumvented as a study [FRAME-BUSTING] demonstrated.

Short of configuring the browser to disable frames and script entirely, which massively impairs browser utility, browser users are vulnerable to this type of attack.

"X-Frame-Options" allows a web page from host B to declare that its content (for example a button, links, text, etc.) must not be displayed in a frame (<frame> or <iframe>) of another page (e.g. from host A). This is done by a policy declared in the HTTP header and enforced by browser implementations as documented here.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. X-Frame-Options Header

The X-Frame-Options HTTP response header field indicates a policy on whether the browser should render the transmitted resource within a <frame> or <iframe>. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, and by this ensuring that their content is not embedded into other pages or frames.

2.1. Syntax

The header field name is:
X-Frame-Options

There are three different values for the header field. These values are mutually exclusive, that is exactly one of the three values MUST be set.

DENY
: A browser receiving content with this header MUST NOT display this content in any frame.

SAMEORIGIN
: A browser receiving content with this header field MUST NOT display this content in any frame from a page of different origin than the content itself.
If a browser or plugin can not reliably determine whether the origin of the content and the frame have the same origin, this MUST be treated as "DENY".
Please note that current implementations vary on the interpretation of this criteria: In some it only allows a page to be framed if the origin of the top-level browsing-context is identical to the origin of the content using the X-FRAME-OPTIONS directive; in others it may consider the origin of the framing page instead. See also section 2.3.2.2 for more details on the nesting of frames and variations in the handling of this header field by different browsers. And refer to section 5 paragraph 2 for the resulting potential security problems.

ALLOW-FROM (followed by a serialized-origin [RFC6454])
: A browser receiving content with this header MUST NOT display this content in a frame from any page with a top-level browsing context of different origin than the specified origin. While this can expose the page to risks by the trusted origin, in some cases it may be necessary to allow the framing by content from other domains.

The meaning of the term "serialized-origin" is given in [RFC6454]. If the ALLOW-FROM value is used, it MUST be followed by a valid origin [RFC6454] (as a subset of URI [RFC3986])
Any data beyond the domain address (i.e. any data after the "/" separator) is to be ignored. And the algorithm to compare origins from [RFC6454] SHOULD be used to verify that a referring page is of the same origin as the content (in the case of SAMEORIGIN) or that the referring page's origin is identical with the ALLOW-FROM serialized-origin (in the case of ALLOW-FROM). Though in conflict with [RFC6454], current implementations do not consider the port as a defining component of the origin. I.e. existing implementations differ with [RFC6454] in that origins with the same protocol but different port values are considered equivalent.

Wildcards or lists to declare multiple domains in one ALLOW-FROM statement are not permitted (see Section 2.3.2.3).

2.2. Augmented Backus-Naur Form (ABNF)

The RFC 5234 [RFC5234] ABNF of the X-Frame-Options header field value is the following.

```
X-Frame-Options = "DENY"
          / "SAMEORIGIN"
          / ( "ALLOW-FROM" RWS SERIALIZED-ORIGIN )

RWS             = 1*( SP / HTAB )
                ; required whitespace
```

With serialized-origin as defined in [RFC6454] and the definition of RWS (required whitespace) is the same as in [HTTPbis-P1].

RWS is used when at least one linear whitespace octet is required to separate field tokens. RWS SHOULD be generated as a single space (SP). Multiple RWS octets that occur within field-content SHOULD either be replaced with a single SP or transformed to all SP octets before interpreting the field value or forwarding the message downstream.

And SP (space) and HTAB (horizontal tab) are as defined in RFC 5234 [RFC5234], Appendix B.1.

The values are specified as ABNF strings, and therefore are case-insensitive.

2.2.1. Examples of X-Frame-Options

```
X-FRAME-OPTIONS: DENY

X-FRAME-OPTIONS: SAMEORIGIN

X-FRAME-OPTIONS: ALLOW-FROM https://example.com/
```

2.3. Design Issues

2.3.1. Enable HTML content from other domains

There are a number of main direct vectors that enable HTML content from other domains and browser implementations of X-Frame-Options cover all of them:

- IFRAME tag
- Frame tag
- The Object tag (requires a redirect)
- Applet tag
- Embed tag

Besides these, other ways to host HTML content can be possible. For example some plugins may host HTML views directly. If these plugins appear essentially as frames (as opposed to top-level windows), the plugins must conform to the X-FRAME-OPTIONS policy as specified in this document as well.

2.3.2. Browser Behaviour and Processing

To allow secure implementations, browsers must behave in a consistent and reliable way.

If an X-Frame-Options HTTP header field prohibits framing, the user-agent of the browser MAY immediately abort downloading or parsing of the document.

2.3.2.1. Violation of X-Frame-Options

When a browser discovers that loaded content with the X-FRAME-OPTIONS header field would be displayed in a frame against the specified orders of the header, the browser SHOULD redirect as soon as possible to a "No-Frame" page. For example this can be a noframe.html page that also states the full URL and hostname of the protected page.

The NoFrame page could provide the user with an option to open the target URL in a new window.

Implementations of this vary, some browsers will show a message that allows the user to safely open the target page in a new window. Other implementations will simply render an empty frame.

2.3.2.2. Variation in current browser behaviour

There are currently variations in the implementation of the X-FRAME-OPTIONS header. For example not all browsers support the "ALLOW-FROM" option. "ALLOW-FROM" was initially an Internet Explorer extension and at the time of writing has not been uniformly implemented by other user agents.

Furthermore the criteria for the SAMEORIGIN (and ALLOW-FROM) directive may not be evaluated unanimously either: The known implementations in Appendix A evaluate the SAMEORIGIN directive based on the origin of the framed page and the top-level browsing-context, while other implementations might evaluate based on the framed page and the framing page, or the whole chain of nested frames inbetween.

To illustrate the difference between the comparison with "framing page" and the "top-level browsing-context" consider the following scenario: Web pages may embed frames with other pages which in turn embed frames with other pages as well and so on. In theory this can result in an infinite nesting of framed pages. For example web page A may contain in a frame web page B, and web page B contains in a frame web page C.

```
Web page A
<html>
....
<frame src="https://URI_of_web_page_B" />
</html>

Web Page B
<html>
....
<frame src="https://URI_of_web_page_C" />
</html>

And so forth...
```

In this example, for the nested frames with the inner framed web page C, the most outer web page A would be the "top-level browsing-context" and web page B would be the "framing page"

These potential variations in the evaluation of the header by different implementations impair the useage and reliability of this http header and have security implications as described in section 5. A revised version of x-frame-options in the form of a frame-options directive in the CSP 1.1[CSP-1-1] will unify the behaviour and it is expected that newer implementations will use it rather than the mechanisms documented here.

2.3.2.3. Usage design pattern and example scenario for the ALLOW-FROM parameter

As the "ALLOW-FROM" field only supports one serialized-origin, in cases when the server wishes to allow more than one resource to frame its content, the following design pattern can fulfil that need:

1. A page that wants to render the requested content in a frame supplies its own origin information to the server providing the to-be-framed content via a querystring parameter.

2. The Server verifies the hostname meets its criteria so that the page can be allowed to be framed by the target resource. This may for example happen via a look-up of a white-list of trusted domain names that are allowed to frame the page. For example, for a Facebook "Like" button, the server can check to see that the supplied hostname matches the hostname(s) expected for that "Like" button.

3. The server returns the hostname in X-FRAME-OPTIONS: ALLOW-FROM if the proper criteria was met in step #2.

4. The browser enforces the X-FRAME-OPTIONS: ALLOW-FROM header.

2.3.2.4. No caching of the X-Frame-Options header

It is not recommended to cache the X-Frame-Options header for a resource. Caching the X-Frame-Options response could result in problems because:

1. The browser has to check for every http-request of the resource whether the X-Frame-Options header has been set and then act accordingly, as a resource itself might be created dynamically and the header could change with it, too.

2. And also, as outlined in section 2.3.2.3., servers may generate X-Frame-Options header responses depending on the request. Example case: Considering that we have only one serialized-origin in the ALLOW-FROM directive, imagine a user has multiple pages open in his browser tabs with one of web page 1 from domain A and the second of web page 2 from domain B, both frame the same page from domain C with the ALLOW-FROM directive. In that case the page needs to reply to both requests with different X-Frame-Options headers, the first pointing to origin A, the second to origin B.

However, we found that none of the major browsers listed in Appendix A cache the responses.

3. Acknowledgements

This document was derived from input from specifications published by various browser vendors such as Microsoft (Eric Lawrence, David Ross), Mozilla, Google, Opera and Apple.

4. IANA Considerations

This memo is a request to IANA to include the specified HTTP header in the registry as outlined in Registration Procedures for Message Header Fields [RFC3864]

4.1. Registration Template

PERMANENT MESSAGE HEADER FIELD REGISTRATION TEMPLATE:

Header field name: X-Frame-Options

Applicable protocol: http [RFC2616]

Status: informational

Author/Change controller: IETF

Specification document(s): draft-ietf-websec-x-frame-options

Related information:

Figure 1

5. Security Considerations

The introduction of the X-FRAME-OPTIONS http header field does improve the protection against Clickjacking. However, it is not self-sufficient on its own to protect against all kinds of these attack vectors. It must be used in conjunction with other security measures like secure coding (e.g. input validation, output encoding, ...) and the Content Security Policy [CSP].

It is important to note that current implementations do not check the origins of the entire ancestor tree of frames of the framing resources, and this may expose the resource to attack in multiple-nested scenarios.

The browser implementations evaluate based on the origin of the framed page and the top-level browsing-context (i.e. most outer frame):

If a resource from origin A embeds untrusted content from origin B, that untrusted content can embed another resource from origin A with an X-Frame-Options: SAMEORIGIN policy and that check would pass when the user agent only verifies the top-level browsing context. Therefore web developers should be aware that embedding content from other sites can leave their web pages vulnerable to clickjacking even if the X-Frame-Options header is used.

Furthermore, X-Frame-Options must be sent as an HTTP header field and is explicitly ignored by user agents when declared with a meta http-equiv tag.

5.1. Privacy Considerations

There are two kinds of potential data leakage to consider:

1. Using X-FRAME-OPTIONS with the parameter ALLOW-FROM allows a page to guess or infer information about who is framing it. A web server may answer requests with the X-FRAME-OPTIONS ALLOW-FROM header and by thus determine which other page is framing it. This is inherent by design, but may lead to data leakage or data protection concerns.

2. The web server using the ALLOW-FROM directive may disclose to other parties who request the page in the header by which page it is allowed to be framed. If a web server wishes to reduce this leakage, it is recommended to generate the ALLOW-FROM header for each request based on the design pattern as described in section 2.3.2.3.

6. References

6.1. 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, January 2005.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

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

6.2. Informative References

[CLICK-DEFENSE-BLOG]
Microsoft, "Clickjacking Defense", 2009, <http://blogs.msdn.com/b/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx>.

[CSP-1-1] Barth, A. and M. West, "Content Security Policy 1.1", W3C Working Draft WD-CSP11-20130604, June 2013, <http://www.w3.org/TR/2013/WD-CSP11-20130604/>.

Latest version available at

[CSP] Sterne, B. and A. Barth, "Content Security Policy 1.0", W3C Candidate Recommendation CR-CSP-20121115, November 2012, <http://www.w3.org/TR/2012/CR-CSP-20121115/>.

Latest version available at

[CSRF] OWASP (Open Web Application Security Project), "OWASP Top-10: Cross-Site Request Forgery (CSRF)", 2010, <https://www.owasp.org/index.php/Top_10_2013-A8-Cross-Site_Request_Forgery_%28CSRF%29>.

[Clickjacking]
OWASP (Open Web Application Security Project), "Clickjacking", 2010, <http://www.owasp.org/index.php/Clickjacking>.

[FRAME-BUSTING]
Stanford Web Security Research, "Busting frame busting: a study of clickjacking vulnerabilities at popular sites", 2010, <http://seclab.stanford.edu/websec/framebusting/>.

[HTTPbis-P1]
IETF, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", 2013, <http://tools.ietf.org/html/draft-ietf-httpbis-p1-messaging-23>.

[Microsoft-X-Frame-Options]
Microsoft, "Combating ClickJacking With X-Frame-Options", 2010, <http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx>.

[Mozilla-X-Frame-Options]
Mozilla, "The X-Frame-Options response header", 2010, <https://developer.mozilla.org/en-US/docs/The_X-FRAME-OPTIONS_response_header>.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

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

[RFC6648] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, June 2012.

Appendix A. Browsers that support X-Frame-Options

- Internet Explorer 8+
- Firefox 3.6.9+
- Opera 10.5+
- Safari 4+
- Chrome 4.1+

Appendix B. Description of a Clickjacking attack

More detailed explanation of Clickjacking scenarios

B.1. Shop

An Internet Marketplace/Shop offering a feature with a link/button to "Buy this" Gadget
The marketplace wants their affiliates (who could be malicious attackers) to be able to stick the "Buy such-and-such from XYZ" IFRAMES into their pages. There is a possible Clickjacking threat here, which is why the marketplace/onlineshop needs to then immediately navigate the main browsing context (or a new window) to a confirmation page which is protected by anti-Clickjacking protections.

B.2. Online Shop Confirm Purchase Page

The "Confirm Purchase" page of an online shop must be shown to the end user without the risk of an overlay or misuse by an attacker. For that reason, the confirmation page uses a combination of anti-CSRF (Cross Site Request Forgery, [CSRF]) tokens and the X-FRAME-OPTIONS HTTP header field, mitigating ClickJacking attacks.

B.3. Flash Configuration

Macromedia Flash configuration settings are set by a Flash object which can run only from a specific configuration page on Macromedia's site. The object runs inside the page and thus can be subject to a ClickJacking attack. In order to prevent ClickJacking attacks against the security settings, the configuration page uses the X -FRAME-OPTIONS directive.

Authors' Addresses

David Ross
Microsoft
U.S.

Tobias Gondrom
Thames Stanley
Kruegerstr. 5A
Unterschleissheim
Germany

Phone: +44 7521003005
Email: tobias.gondrom@gondrom.org