WEBSEC D. Ross
Internet-Draft Microsoft
Intended status: Informational T. Gondrom
Expires: August 29, 2013 Thames Stanley
February 25, 2013

HTTP Header Field X-Frame-Options
draft-ietf-websec-x-frame-options-02

Abstract

To improve the protection of web applications against Clickjacking, this specification 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 August 29, 2013.

Copyright Notice

Copyright (c) 2013 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 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. Requirements Language 3
2. X-Frame-Options Header 3
  2.1. Syntax 4
  2.2. Augmented Backus-Naur Form (ABNF) 5
    2.2.1. Examples of X-Frame-Options 5
    2.2.2. Variations of the ALLOW-FROM field 5
  2.3. Design Issues 5
    2.3.1. Enable HTML content from other domains 5
    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 7
3. Acknowledgements 7
4. IANA Considerations 8
  4.1. Registration Template 8
5. Security Considerations 8
  5.1. Privacy Considreations 9
6. References 9
  6.1. Normative References 9
  6.2. Informative References 9
Appendix A. Browsers that support X-Frame-Options 10
Appendix B. Description of a Clickjacking attack 10
  B.1. Shop 10
  B.2. Online Shop Confirm Purchase Page 11
  B.3. Flash Configuration 11
Authors' Addresses 11

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 [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, possibly belonging to another domain and thereby triggering actions on the second server B without the knowledge nor intention of the user and potentially using an existing session context and login in that step.

This specification provides informational documentation about the current use and definition of the X-Frame-Options HTTP header field. 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 secure 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). In principle this is done by a policy declared in the HTTP header and enforced by conforming browser implementations.

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 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 compare to the origin of the framing page.

ALLOW-FROM (followed by a URI [RFC3986] of a trusted origin)

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.

If the ALLOW-FROM value is used, it MUST be followed by a valid URI. 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 or that the referring page's origin is identical with the ALLOW-FROM URI. Though in conflict with [RFC6454], current implementations do not consider the port as a

defining component of the origin.

Wildcards or lists to declare multiple domains in one ALLOW-FROM statement are not permitted.

2.2. Augmented Backus-Naur Form (ABNF)

The RFC 2234 [RFC2234] ABNF of the X-Frame-Options header is:

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

With URI as defined in [RFC3986] and RWS and OWS as defined in [HTTPbis-P1]. 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/
```

2.2.2. Variations of the ALLOW-FROM field

Regarding the syntax, it should be noted that existing implementations have variations in whether a colon (":") should be between "ALLOW-FROM" and the URI. E.g. in IE8+ the colon ":" is not needed, while Firefox and Chrome implementations at the time of writing of this document support both forms.

Alternative ABNF of the X-Frame-Options header:

```
        X-Frame-Options = "DENY"
                / "SAMEORIGIN"
                / ( "ALLOW-FROM" OWS ":" OWS URI )
```

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:

- 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.

"No-Frame" Page
If the display of content is denied by the X-FRAME-OPTIONS header an error page SHOULD be displayed. For example this can be a noframe.html page also stating the full URL of the protected page and the hostname of the protected page.

The NoFrame page MAY 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 IE (Internet Explorer)

extension and at the time of writing has not been uniformly implemented by other user agents.

And the criteria for the SAMEORIGIN option is not evaluated unanimously either: one implementation may evaluate the SAMEORIGIN option based on the origin of the framed page and the framing page, while another may evaluate based on the framed page and the top-level browsing-context.

These variations in the evaluation of the header by different implementations impair the useage and reliability of this http header. A revised version of x-frame-options in the form of a frame-options directive in the CSP 1.1[CSP-1-1] shall unify the behaviour and replace this document in the future.

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

As the "ALLOW-FROM" field does support only one URI, in cases when the server wishes to allow more than one resource to frame its content, the following design pattern is recommended:

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 return 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.

3. Acknowledgements

This document was derived from input from specifications published by various browser vendors like 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-Option

Applicable protocol: http [RFC2616]

Status: Standard

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, but must be used in conjunction with other security measures like secure coding 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 multiply-nested scenarios. For example, if a resource on 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 if the user agent only verifies the top-level browsing context.

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 Considreations

The parameter ALLOW-FROM allows a page to guess who is framing it. This is inherent by design, but may lead to data leakage or data protection concerns.

6. References

6.1. Normative References

[HTTPbis-P1]
IETF, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", 2013, <http://tools.ietf.org/html/draft-ietf-httpbis-p1-messaging-22>.

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

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[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] W3C, "Content Security Policy 1.0", November 2012, <http://www.w3.org/TR/CSP/>.

[CSP-1-1] W3C, "Content Security Policy 1.1", December 2012, <http://www.w3.org/TR/CSP11/>.

[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/>.

[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 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.

Phone:
Email:

Tobias Gondrom
Thames Stanley
Kruegerstr. 5A
Unterschleissheim,
Germany

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