WEBSEC
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2012

D. Ross
Microsoft
T. Gondrom

March 5, 2012

HTTP Header Frame Options
draft-gondrom-frame-options-02

Abstract

To improve the protection of web applications against Cross Site Request Forgery (CSRF) and Clickjacking this standards defines a http response header that declares a policy communicated from a host to the client browser whether the transmitted content MUST NOT be displayed in frames of other pages from different origins or a list of trusted origins which are allowed to frame the content.

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 6, 2012.

Table of Contents

1. Introduction

In 2009 and 2010 many browser vendors introduced the use of a non-standard http header RFC 2616 [RFC2616] "X-Frame-Options" to protect against Clickjacking [Clickjacking] and Cross Site Request Forgery (CSRF) [CSRF]. This standard is to replace the non-standard header.

In some forms of Clickjacking and CSRF an attacker tricks a user into clicking on a button or link to another page and by thus executing an unintended command in the context of a different web application. For example with Clickjacking the attacker might use transparent or opaque layers to integrate and obscure a button to another page so that the user may click on it in the expectation of a different action. So, in this way the attacker is "hijacking" the "Click" on a button meant by the user to be sent to host A, while clicking the button in effect sends a message to host B. If the user does for example also have an open session with host B this can lead to a CSRF attack and executing a command in the session context of the user (using the user's authentication and authorization) on host B without his intention or knowledge.

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.

The by "Frame-Options" provided defense mechanism against Clickjacking is to allow 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 of another page (e.g. from host A). In principle this is done by a policy declared in the HTTP header and obeyed by conform 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. Frame-Options Header

The Frame-Options HTTP response header indicates a policy whether a browser MUST NOT allow to render a page in a <frame> or <iframe> . Hosts can declare this policy in the header of their HTTP responses

to prevent clickjacking attacks, by ensuring that their content is not embedded into other pages or frames.

2.1. Syntax

The header field name is:
Frame-Options

There are three different values for the header field. These values are exclusive, that is NOT more than 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 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".
[TBD]current implementations do not display if the origin of the top-level-browsing-context is different than the origin of the page containing the FRAME-OPTIONS header.

ALLOW-FROM (followed by a list of URIs of trusted origins)
: A browser receiving content with this header MUST NOT display this content in any frame from a page of different origin than any of the listed origins. This allows deployment with multi-domain sites, as the webmaster can define a whitelist of origins that are allowed to frame the page. While this can expose the page to risks by the trusted origins, in some cases it may be necessary to use content from other domains or more than one origin (hostname).
for example: FRAME-OPTIONS: ALLOW-FROM https://www.domain.com/

In the case of SAMEORIGIN and ALLOW-FROM, there is also an optional flag "AllAncestors". If this flag is set, it means that browsers MUST validate the URL of each hosting frame up to the top level and only allow the framing if all ancestor frames' origins are either the same as in SAMEORIGIN or included in the ALLOW-FROM list.

The URIs listed for ALLOW-FROM must be valid.
Any data beyond the domain address (i.e. any data after the "/" separator) is to be ignored and to verify a referring page is of the same origin as the content or that the referring page is listed in

the ALLOW-FROM list of URI, the algorithm to compare origins from [ORIGIN] should be used.

Wildcards to declare multiple domains in one statement are not permitted.

[TBD] Current Implementations do not consider the port a component of the origin - conflicting with [ORIGIN].

2.2. Backus-Naur Form (BNF)

The RFC 822 [RFC0822] EBNF of the Frame-Options header is:

```
      Frame-Options = "Frame-Options" ":" "DENY"/ "SAMEORIGIN" /
                              ("ALLOW-FROM" ":" Origin-List) : flags
          Origin-List = 1*URI

          flags    = token [ "=" ( token | quoted-string ) ]
```

[TBD] with URI as defined in the websec-origin draft
[TBD] Or should we use the ABNF (RFC 2234) alternatively or in addition?

2.3. Design Issues

2.3.1. Enable HTML content from other domains

There are three main direct vectors that enable HTML content from other domains:

- IFRAME Tag
- Frame tag
- The Object tag (requires a redirect)

Besides these other ways to host HTML content can be possible. For example some plugins may host HTML views directly. To allow a conform security configuration those plugins MUST be conform to the FRAME-OPTIONS directive as specified in this draft as well.

2.3.2. Browser Behaviour and Processing

To allow secure implementations browser implementations MUST behave in a consistent and reliable way conform to thsi specition.

If a HTTP Header prohibits framing, the user-agent of the browser MAY

immediately abort downloading or parsing of the document.

When a browser discovers loaded content with the FRAME-OPTIONS header would be displayed in a frame against the specified origin 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 FRAME-OPTIONS header an accroding 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.

[TBD] The NoFrame page MAY provide the user with an option to open the target URL in a new window.

2.4. Examples of Frame-Options Headers

2.4.1. Example scenario for the ALLOW-FROM parameter

1. Inner IFRAME suggests via a querystring parameter what site it wants to be hosted by. This can obviously be specified by an attacker, but that's OK.

2. Server verifies the hostname meets whatever criteria. For example, for a Facebook "Like" button, the server can check to see that the supplied hostname matches the hostname expected for that Like button.

3. Server serves up the hostname in X-FRAME-OPTIONS: ALLOW-FROM if the proper criteria was met in step #2.

4. Browser enforces the X-FRAME-OPTIONS: ALLOW-FROM domain.com 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 a request to IANA to include the specified HTTP header in registry as outlined in Registration Procedures for Message Header Fields [RFC3864]

4.1. Registration Template

PERMANENT MESSAGE HEADER FIELD REGISTRATION TEMPLATE:

Header field name: Frame-Option

Applicable protocol: http [RFC2616]

Status: Standard

Author/Change controller: IETF

Specification document(s): draft-gondrom-frame-options

Related information:

Figure 1

5. Security Considerations

The introduction of the http header FRAME-OPTIONS 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 Content Security Policy (CSP)

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

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.

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>.

[CSRF] OWASP (Open Web Application Security Project), "OWASP Top-10: Cross-Site Request Forgery (CSRF)", 2010,

<http://www.owasp.org/index.php/Top_10_2010-A5>.

[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/>.

[ORIGIN] IETF, "The Web Origin Concept", December 2010, <http://tools.ietf.org/id/draft-ietf-websec-origin-00.txt>.

[RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[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.

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

Appendix A. Description of a Clickjacking attack

More detailed explanation of Clickjacking scenarios

A.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 bad guys) to be able to stick the "Buy such-and-such from XYZ" IFRAMES into their pages. There is a CSRF-ClickJack possibility 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-CSRF/anti-CJ protections.

A.2. Confirm Purchase Page

Onlineshop "Confirm purchase" anti-CSRF page
The Confirm Purchase page must be shown to the end user without possibility of overlay or misuse by an attacker. For that reason, the confirmation page uses anti-CSRF tokens and contains the FRAME-OPTIONS directive, mitigating ClickJack attacks.

A.3. Flash Configuration

Macromedia Flash configuration page
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 FRAME-OPTIONS directive.

Authors' Addresses

David Ross
Microsoft
U.S.

Phone:
Email:

Tobias Gondrom
Kruegerstr. 5A
Unterschleissheim,
Germany

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