Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 28, 2017

M. Nottingham
November 24, 2016

Site-Wide HTTP Headers
draft-nottingham-site-wide-headers-01

Abstract

This document specifies an alternative way for Web sites to send HTTP response header fields that apply to an entire origin, to improve efficiency.

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 May 28, 2017.

Table of Contents

1. Introduction

HTTP response headers are being used for an increasing amount of metadata that applies to an entire Web site (i.e., the entire origin, as per [RFC6454]).

For example, "Strict-Transport-Security" [RFC6797] and "Public-Key-Pins" [RFC7469] both define headers that are explicitly scoped to an entire origin, and number of similar headers are under consideration.

Likewise, some HTTP header fields only sensibly have a single value per origin; for example, "Server".

Furthermore, some headers are used uniformly across an origin. For example, a site might have a homogenous "Content-Security-Policy" [W3C.CR-CSP2-20150721] header.

HTTP/2's HPACK [RFC7541] header compression mechanism was designed to reduce bandwidth usage for often-repeated headers, both in responses and requests. However, it limits the amount of compression contents usable for a connection (by default, 4K), and some sites are beginning to exceed this limit, thereby reducing the efficiency of HPACK itself.

For example, it is not uncommon for a CSP response header field to exceed 1K (and has been observed to be greater than 3K on popular sites). This forces site administrators to make an awkward choice;

put the large header in the HPACK table, thereby crowding out other headers, or omit it, requiring its full content to be sent on every applicable response.

This document defines a way to specify one or more sets of HTTP response header fields in a well-known resource [RFC5785] that, when their use is negotiated, are appended to the header blocks of all HTTP responses on that site by the user agent. This allows common response headers to be omitted both from on-the-wire responses and the HPACK compression table, making both more efficient.

This approach is preferable to increasing the HTTP/2 SETTINGS_HEADER_TABLE_SIZE ([RFC7540], Section 6.5.2), because increasing that setting incurs a per-connection overhead on the server, whereas using the technique documented here does not.

1.1. Selecting Site-Wide Headers

Only certain header fields are suitable for being set for an entire origin. Therefore, a header field MUST be listed below, or its field name MUST start with the characters "site-" (case insensitive) to be usable as a site-wide header.

The whitelisted field names are:

- Access-Control-Allow-Origin
- Alt-Svc
- Content-Security-Policy
- P3P
- Public-Key-Pins
- Public-Key-Pins-Report-Only
- Server
- Strict-Transport-Security

Note that inclusion in this list does not imply that a header field is always site-wide.

Future specifications SHOULD NOT update this whitelist; instead, they SHOULD use the "site-" prefix.

1.2. Example

If a user agent has a fresh copy of the well-known resource for an origin (see Section 4) (e.g., because it performed a GET, or HTTP/2 Server Push was used):

```
HTTP/1.1 200 OK
Content-Type: text/site-headers
Cache-Control: max-age=3600
ETag: "abc123"
Content-Length: 284

Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
Site-Foo: bar
```

and the user agent makes a subsequent request:

```
GET /images/foo.jpg HTTP/1.1
Host: www.example.com
SH: "abc123"
```

That indicates that the user agent has processed the well-known resource (because the "SH" header field is present, and its value matches the current value of the "ETag" of the well-known resource). Therefore, the server can omit the nominated response header fields on the wire, replacing them with the "HS" response header field, whose value is the same as that of the "SH" field:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Vary: SH, Accept-Encoding
Cache-Control: max-age=3600
Transfer-Encoding: chunked
HS: "abc123"
```

Upon receipt of that response, the user agent will consider it equivalent to:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Vary: SH, Accept-Encoding
Cache-Control: max-age=3600
Connection: close
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
Site-Foo: bar
```

If a request omits the "SH" header field, or its field-value does not match the current "ETag" of the well-known resource, all of the header fields above will be sent by the server in the response, and "HS" will not be sent.

1.3. Notational Conventions

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].

This document uses the following ABNF rules from [RFC7230]: "OWS", "CRLF", "header-field". From [RFC7232]: "entity-tag".

2. Server Operation

When a server wishes to use site-wide HTTP headers, it places a file in the format specified in Section 4.1 at the well-known URI specified in Section 4. That file SHOULD NOT contain header fields not allowed by Section 1.1.

Then, when a request has a "SH" request header field (as per Section 3.1) whose value matches the current ETag of the well-known resource, the set of response header fields in the payload of the well-known resource are omitted from the corresponding response, and the "HS" response header field is sent with the same value as the "SH" request header field.

Servers MUST include "SH" in the field-value of the "Vary" response header field for all cacheable (as per [RFC7234]) responses of resources that behave in this manner, whether or not headers have been actually appended. This assures correct cache operation, and also advertises support for this specification.

Servers MAY use HTTP/2 Server Push ([RFC7540], Section 8.2) to proactively send the well-known resource to user agents (e.g., if they emit "SH: *", indicating that they do not have a fresh copy of the well-known resource).

3. User Agent Operation

User agents that support this specification SHOULD always emit a "SH" header field in requests.

When a valid representation of the well-known resource is held (as defined in Section 4), its value will be its "ETag". When one is not (e.g., because it has not been requested, the one held is syntactically invalid, or it is stale, as per [RFC7234]), its value is "*" (unquoted).

When an "ETag" is sent and the response contains the "HS" response header field (see Section 3.2), user agents MUST confirm that the value of the "HS" response header is character-for-character identical (after removing leading and trailing whitespace) to that of the "SH" request header field it sent. If it is not, the response MUST be considered invalid and MUST NOT be used; the user agent MAY retry the request without the "SH" request header field if its method was safe, MAY attempt to re-fetch the well-known location beforehand, and MAY take alternative recovery strategies.

If the values match, the user agent MUST append the contents of the well-known resource that are currently held to be appended to the response headers received, but MUST NOT include any headers not allowed by Section 1.1.

3.1. The "SH" HTTP Request Header Field

The "SH" HTTP request header field indicates that the user agent has a fresh (as per [RFC7234]) copy of the well-known resource (see Section 4) for the request's origin ([RFC6454]).

SH = "*" / entity-tag

Its value is the "entity-tag" [RFC7232] of the freshest valid well-known location response held by the user agent. If none is held, it should be "*" (without quotes).

For example:

```
SH: "abc123"
SH: *
```

3.2. The "HS" HTTP Response Header Field

The "HS" HTTP response header field indicates that the server has chosen to omit the headers in the well-known resource's response that shares its "ETag" with the field value.

HS = entity-tag

Its value is the "entity-tag" [RFC7232] of the well-known response whose headers are being used, and MUST match that received in the "SH" header field of the request.

For example:

HS: "abc123"

4. The "site-headers" well-known URI

The well-known URI [RFC5785] "site-headers" is a resource that, when fetched, returns a representation in the "text/site-headers" format (see Section 4.1).

Its media type SHOULD be generated as "text/site-headers", although user agents SHOULD NOT reject responses with other types (particularly, "application/octet-stream" and "text/plain").

Its representation MUST contain an "ETag" response header [RFC7232].

User agents SHOULD NOT consider it valid if it fails to parse, but MAY attempt to recover from errors in a manner similar to how headers are normally handled.

User agents SHOULD consider it to be valid for its freshness lifetime (as per [RFC7234]). If it does not have an explicit freshness lifetime, they SHOULD consider it to have a heuristic freshness lifetime of 120 seconds.

4.1. The "text/site-headers" Media Type

The "text/site-headers" media type is used to indicate that a file contains a set of HTTP header fields, as defined in [RFC7230], Section 3.

site-headers = OWS *(header-field CRLF) OWS

As in HTTP itself, implementations need to be forgiving about line endings; specifically, bare CR MUST be considered to be a line ending.

For example:

```
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
Content-Security-Policy: default-src 'self'; img-src 'self'
  *.staticflickr.com; frame-ancestors 'none';
  report-uri https://mnot.report-uri.io/r/default/csp/enforce
```

Note that the "Public-Key-Pins" and "Content-Security-Policy" header fields are line-folded; as in HTTP, this form of header is deprecated in this format, and SHOULD NOT be used (except in documentation, as we see here).

5. IANA Considerations

TBD

6. Security Considerations

6.1. Injecting Headers

Site-wide headers allow a single resource to inject HTTP response headers for an entire origin. Accordingly, the ability to write to that resource needs to be carefully controlled by the origin server.

6.2. Differing Views of Headers

Because headers sent via this mechanism will not be seen by user agents and intermediaries that do not implement this specification, they will potentially have a different view of the response headers.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <http://www.rfc-editor.org/info/rfc5785>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <http://www.rfc-editor.org/info/rfc6454>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <http://www.rfc-editor.org/info/rfc7232>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <http://www.rfc-editor.org/info/rfc7234>.

7.2. Informative References

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <http://www.rfc-editor.org/info/rfc6797>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <http://www.rfc-editor.org/info/rfc7469>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <http://www.rfc-editor.org/info/rfc7540>.

[RFC7541] Peon, R. and H. Ruellan, "HPACK: Header Compression for HTTP/2", RFC 7541, DOI 10.17487/RFC7541, May 2015, <http://www.rfc-editor.org/info/rfc7541>.

[W3C.CR-CSP2-20150721] West, M., Barth, A., and D. Veditz, "Content Security Policy Level 2", World Wide Web Consortium CR CR-CSP2-20150721, July 2015, <http://www.w3.org/TR/2015/CR-CSP2-20150721>.

Author's Address

Mark Nottingham

Email: mnot@mnot.net
URI: https://www.mnot.net/