HTTP Working Group M. Nottingham
Internet-Draft
Intended status: Standards Track E. Nygren
Expires: October 22, 2017 Akamai
April 20, 2017

The ORIGIN HTTP/2 Frame
draft-ietf-httpbis-origin-frame-03

Abstract

This document specifies the ORIGIN frame for HTTP/2, to indicate what origins are available on a given connection.

Note to Readers

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/ .

Working Group information can be found at http://httpwg.github.io/ ; source code and issues list for this draft can be found at https://github.com/httpwg/http-extensions/labels/origin-frame .

Status of This Memo

This Internet-Draft will expire on October 22, 2017.

Copyright Notice

Table of Contents

1. Introduction

HTTP/2 [RFC7540] allows clients to coalesce different origins [RFC6454] onto the same connection when certain conditions are met. However, in certain cases, a connection is is not usable for a coalesced origin, so the 421 (Misdirected Request) status code ([RFC7540], Section 9.1.2) was defined.

Using a status code in this manner allows clients to recover from misdirected requests, but at the penalty of adding latency. To address that, this specification defines a new HTTP/2 frame type, "ORIGIN", to allow servers to indicate what origins a connection is usable for.

Additionally, experience has shown that HTTP/2's requirement to establish server authority using both DNS and the server's certificate is onerous. This specification relaxes the requirement to check DNS when the ORIGIN frame is in use. Doing so has additional benefits, such as removing the latency associated with some DNS lookups, and improving DNS privacy.

1.1. 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].

2. The ORIGIN HTTP/2 Frame

The ORIGIN HTTP/2 frame ([RFC7540], Section 4) allows a server to indicate what origin(s) [RFC6454] the server would like the client to consider as members of the Origin Set (Section 2.3) for the connection it occurs within.

2.1. Syntax

The ORIGIN frame type is 0xc (decimal 12).

```
+-------------------------------+-------------------------------+
|         Origin-Len (16)       | ASCII-Origin? (*)           ...
+-------------------------------+-------------------------------+
```

The ORIGIN frame's payload contains the following fields, sets of which may be repeated within the frame to indicate multiple origins:

Origin-Len: An unsigned, 16-bit integer indicating the length, in octets, of the ASCII-Origin field.

Origin: An optional sequence of characters containing the ASCII serialization of an origin ([RFC6454], Section 6.2) that the sender believes this connection is or could be authoritative for.

The ORIGIN frame does not define any flags. However, future updates to this specification MAY define flags. See Section 2.2.

2.2. Processing ORIGIN Frames

The ORIGIN frame is a non-critical extension to HTTP/2. Endpoints that do not support this frame can safely ignore it upon receipt.

When received by an implementing client, it is used to initialise and manipulate the Origin Set (see Section 2.3), thereby changing how the client establishes authority for origin servers (see Section 2.4).

The origin frame MUST be sent on stream 0; an ORIGIN frame on any other stream is invalid and MUST be ignored.

Likewise, the ORIGIN frame is only valid on connections with the "h2" protocol identifier, or when specifically nominated by the protocol's definition; it MUST be ignored when received on a connection with the "h2c" protocol identifier.

This specification does not define any flags for the ORIGIN frame, but future updates might use them to change its semantics. The first four flags (0x1, 0x2, 0x4 and 0x8) are reserved for backwards-incompatible changes, and therefore when any of them are set, the ORIGIN frame containing them MUST be ignored by clients conforming to this specification, unless the flag's semantics are understood. The remaining flags are reserved for backwards-compatible changes, and do not affect processing by clients conformant to this specification.

The ORIGIN frame describes a property of the connection, and therefore is processed hop-by-hop. An intermediary MUST NOT forward ORIGIN frames. Clients configured to use a proxy MUST ignore any ORIGIN frames received from it.

Each ASCII-Origin field in the frame's payload MUST be parsed as an ASCII serialisation of an origin ([RFC6454], Section 6.2). If parsing fails, the field MUST be ignored.

See Appendix A for an illustrative algorithm for processing ORIGIN frames.

2.3. The Origin Set

The set of origins (as per [RFC6454]) that a given connection might be used for is known in this specification as the Origin Set.

By default, a connections's Origin Set is uninitialised. When an ORIGIN frame is first received and successfully processed by a client, the connection's Origin Set is defined to contain a single origin, composed from:

- Scheme: "https"

- Host: the value sent in Server Name Indication ([RFC6066] Section 3), converted to lower case

- Port: the remote port of the connection (i.e., the server's port)

The contents of that ORIGIN frame (and subsequent ones) allows the server to incrementally add new origins to the Origin Set, as described in Section 2.2.

The Origin Set is also affected by the 421 (Misdirected Request) response status code, defined in [RFC7540] Section 9.1.2. Upon receipt of a response with this status code, implementing clients

MUST create the ASCII serialisation of the corresponding request's origin (as per [RFC6454], Section 6.2) and remove it from the connection's Origin Set, if present.

2.4. Authority, Push and Coalescing with ORIGIN

[RFC7540], Section 10.1 uses both DNS and the presented TLS certificate to establish the origin server(s) that a connection is authoritative for, just as HTTP/1.1 does in [RFC7230]. Furthermore, [RFC7540] Section 9.1.1 explicitly allows a connection to be used for more than one origin server, if it is authoritative. This affects what requests can be sent on the connection, both in HEADERS frame by the client and as PUSH_PROMISE frames from the server.

Once an Origin Set has been initialised for a connection, clients that implement this specification change these behaviors in the following ways:

- Clients MUST NOT consult DNS to establish the connection's authority for new requests. The TLS certificate MUST stil be used to do so, as described in [RFC7540] Section 9.1.1.

- Clients sending a new request SHOULD use an existing connection if the request's origin is in that connection's Origin Set, unless there are operational reasons for creating a new connection.

- Clients MUST use the Origin Set to determine whether a received PUSH_PROMISE is authoritative, as described in [RFC7540], Section 8.2.2.

Note that clients are still required to perform checks on the certificate presented by the server for each origin that a connection is used for; see [RFC7540] Section 9.1.1 for more information. This includes verifying that the host matches a "dNSName" value from the certificate "subjectAltName" field (using the wildcard rules defined in [RFC2818]; see also [RFC5280] Section 4.2.1.6).

Because ORIGIN can change the set of origins a connection is used for over time, it is possible that a client might have more than one viable connection to an origin open at any time. When this occurs, clients SHOULD not emit new requests on any connection whose Origin Set is a subset of another connection's Origin Set, and SHOULD close it once all outstanding requests are satisfied.

3. IANA Considerations

This specification adds an entry to the "HTTP/2 Frame Type" registry.

- Frame Type: ORIGIN

- Code: 0xc

- Specification: [this document]

4. Security Considerations

Clients that blindly trust the ORIGIN frame's contents will be vulnerable to a large number of attacks. See Section 2.4 for mitigations.

Relaxing the requirement to consult DNS when determining authority for an origin means that an attacker who possesses a valid certificate no longer needs to be on-path to redirect traffic to them; instead of modifying DNS, they need only convince the user to visit another Web site, in order to coalesce connections to the target onto their existing connection.

5. References

5.1. 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>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <http://www.rfc-editor.org/info/rfc6066>.

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

[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>.

5.2. Informative References

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010, <http://www.rfc-editor.org/info/rfc5988>.

[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>.

[RFC7838] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <http://www.rfc-editor.org/info/rfc7838>.

Appendix A. Non-Normative Processing Algorithm

The following algorithm illustrates how a client could handle received ORIGIN frames:

1. If the client is configured to use a proxy for the connection, ignore the frame and stop processing.

2. If the connection is not identified with the "h2" protocol identifier or another protocol that has explicitly opted into this specification, ignore the frame and stop processing.

3. If the frame occurs upon any stream except stream 0, ignore the frame and stop processing.

4. If any of the flags 0x1, 0x2, 0x4 or 0x8 are set, ignore the frame and stop processing.

5. If no previous ORIGIN frame on the connection has reached this step, initialise the Origin Set as per Section 2.3.

6. For each Origin field "origin_raw" in the frame payload:

1. Parse "origin_raw" as an ASCII serialization of an origin ([RFC6454], Section 6.2) and let the result be "parsed_origin". If parsing fails, skip to the next "origin_raw".

2. Add "parsed_origin" to the Origin Set.

Appendix B. Operational Considerations for Servers

The ORIGIN frame allows a server to indicate for which origins a given connection ought be used.

For example, it can be used to inform the client that the connection is to only be used for the SNI-based origin, by sending an empty ORIGIN frame. Or, a larger number of origins can be indicated by including a payload.

Generally, this information is most useful to send before sending any part of a response that might initiate a new connection; for example, "Link" headers [RFC5988] in a response HEADERS, or links in the response body.

Therefore, the ORIGIN frame ought be sent as soon as possible on a connection, ideally before any HEADERS or PUSH_PROMISE frames.

However, if it's desirable to associate a large number of origins with a connection, doing so might introduce end-user perceived latency, due to their size. As a result, it might be necessary to select a "core" set of origins to send initially, expanding the set of origins the connection is used for with subsequent ORIGIN frames later (e.g., when the connection is idle).

That said, senders are encouraged to include as many origins as practical within a single ORIGIN frame; clients need to make decisions about creating connections on the fly, and if the origin set is split across many frames, their behaviour might be suboptimal.

Senders take note that, as per [RFC6454] Section 4, the values in an ORIGIN header need to be case-normalised before serialisation.

Finally, servers that host alternative services [RFC7838] will need to explicitly advertise their origins when sending ORIGIN, because the default contents of the Origin Set (as per Section 2.3) do not contain any Alternative Services' origins, even if they have been used previously on the connection.

Authors' Addresses

Mark Nottingham

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

Erik Nygren
Akamai

Email: nygren@akamai.com