Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 14, 2014

M. Nottingham
Akamai
P. McManus
Mozilla

December 11, 2013

HTTP Alternate Services
draft-nottingham-httpbis-alt-svc-01

Abstract

This document introduces "alternate services" to allow an HTTP origin's resources to be available at a separate network location, possibly accessed with a different protocol configuration.

It also specifies one means of discovering alternate services, the "Alt-Svc" 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 June 14, 2014.

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

[I-D.ietf-httpbis-http2] specifies a few ways to negotiate the use of HTTP/2.0 without changing existing URIs. However, several deficiencies in using the "upgrade dance" for "http://" URIs have become apparent. While that mechanism is still being investigated, some have expressed interest in an alternate approach.

Furthermore, some implementers have expressed a strong desire utilize HTTP/2 only in conjunction with TLS. Alternate-Services provides a potential mechanism for achieving that for "http://" URIs; see [I-D.nottingham-http2-encryption] for details.

Finally, HTTP/2.0 is designed to have longer-lived, fewer and more active TCP connections. While these properties are generally "friendlier" for the network, they can cause problems for servers that currently exploit the short-lived flow characteristics of HTTP/1.x for load balancing, session affinity and maintaining locality to the user.

This document specifies a new concept in HTTP, the "alternate service," to address these use cases. An alternate service can be used to interact with the resources on an origin server at a separate location on the network, possibly using a different protocol configuration.

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].

This document uses the Augmented BNF defined in [RFC5234] along with the "OWS", "DIGIT", "parameter", "uri-host", "port" and "delta-second" rules from [I-D.ietf-httpbis-p1-messaging], and uses the "#rule" extension defined in Section 7 of that document.

2. Alternate Services

On the Web, a resource is accessed through a scheme (e.g., "https" or "http") on a nominated host / port combination.

These three pieces of information collectively can be used to establish the authority for ownership of the resource (its "origin"; see [RFC6454]), as well as providing enough information to bootstrap access to it.

This document introduces the notion of an "Alternate Service"; when an origin's resources are accessible through a different protocol / host / port combination, it is said to have an alternate service.

For example, an origin:

("http", "www.example.com", "80")

might declare that its resources are also accessible at the alternate service:

("http2-tls", "new.example.com", "443")

By their nature, alternate services are explicitly at the granularity of an origin; i.e., they cannot be selectively applied to resources within an origin.

Alternate services do not replace or change the origin for any given resource; in general, they are not visible to the software "above" the access mechanism. The alternate service is essentially alternate routing information that can also be used to reach the origin in the same way that DNS CNAME or SRV records define routing information at the name resolution level.

Furthermore, it is important to note that the first member of an alternate service tuple is different from the "scheme" component of an origin; it is more specific, identifying not only the major version of the protocol being used, but potentially communication options for that protocol.

This means that clients using an alternate service will change the host, port and protocol that they are using to fetch resources, but these changes MUST NOT be propagated to the application that is using HTTP; from that standpoint, the URI being accessed and all information derived from it (scheme, host, port) are the same as before.

Importantly, this includes its security context; in particular, when TLS [RFC5246] is in use, the alternate server will need to present a certificate for the origin's host name, not that of the alternate. Likewise, the Host header is still derived from the origin, not the alternate service (just as it would if a CNAME were being used).

The changes MAY, however, be made visible in debugging tools, consoles, etc.

Formally, an alternate service is identified by the combination of:

- An ALPN protocol, as per [I-D.ietf-tls-applayerprotoneg]
- A host, as per [RFC3986]
- A port, as per [RFC3986]

Additionally, each alternate service MUST have:

- A freshness lifetime, expressed in seconds; see Section 2.1.2
- A numeric priority; see Section 2.1.3

Potentially, there are many ways that a client could discover the alternate service(s) associated with an origin; this document currently defines one, the Alt-Svc HTTP Header Field (Section 3).

2.1. Client Handling for Alternate Services

2.1.1. Host Authentication

Clients MUST NOT use alternate services with a host other than the origin's without strong server authentication; this mitigates the attack described in Section 4.2. One way to achieve this is for the alternate to use TLS with a certificate that is valid for that origin.

For example, if the origin's host is "www.example.com" and an alternate is offered on "other.example.com" with the "http2-tls" protocol, and the certificate offered is valid for "www.example.com", the client can use the alternate. However, if "other.example.com" is offered with the "http2" protocol, the client cannot use it, because there is no mechanism in that protocol to establish strong server authentication.

2.1.2. Alternate Service Caching

Mechanisms for discovering alternate services can associate a freshness lifetime with them; for example, the Alt-Svc header field uses the "ma" parameter.

Clients MAY choose to use an alternate service instead of the origin at any time when it is considered fresh; see Section 2.1.5 for specific recommendations.

To mitigate risks associated with caching compromised values (see Section 4.2 for details), user agents SHOULD examine cached alternate services when they detect a change in network configuration, and remove any that could be compromised (for example, those whose association with the trust root is questionable). UAs that do not have a means of detecting network changes SHOULD place an upper bound on their lifetime.

2.1.3. Alternate Service Priorities

Mechanisms for discovering alternate services can associate a priority with them; for example, the Alt-Svc header field uses the "pr" parameter.

Priorities are numeric, with a range of 1-64, and are relative to the origin server, which has a static priority of 32. Higher values are preferable.

Therefore, an alternate with a priority of 48 will be used in preference to the origin server, whereas one with a priority of 10 will be used only when the origin server becomes unavailable.

Note that priorities are not specific to the mechanism that an alternate was discovered with; i.e., there is only one "pool" of priorities for an origin.

2.1.4. Requiring Server Name Indication

A client must only use a TLS based alternate service if the client also supports TLS Server Name Indication (SNI) [RFC6066]. This supports the conservation of IP addresses on the alternate service host.

2.1.5. Using Alternate Services

By their nature, alternate services are optional; clients are not required to use them. However, it is advantageous for clients to behave in a predictable way when they are used by servers (e.g., for load balancing).

Therefore, if a client becomes aware of an alternate service that has a higher priority than a connection currently in use, the client SHOULD use that alternate service as soon as it is available, provided that the security properties of the alternate service protocol are desirable, as compared to the existing connection.

For example, if an origin advertises a "http2-tls" alternate service using an "Alt-Svc" response header field, the client ought to immediately establish a connection to the most preferable alternate service, and use it in preference to the origin connection once available.

The client is not required to block requests; the origin's connection can be used until the alternate connection is established. However, if the security properties of the existing connection are weak (e.g. cleartext HTTP/1.1) then it might make sense to block until the new

connection is fully available in order to avoid information leakage.

Furthermore, if the connection to the alternate service fails or is unresponsive, the client MAY fall back to using the origin, or a less preferable alternate service.

3. The Alt-Svc HTTP Header Field

A HTTP(S) origin server can advertise the availability of alternate services to clients by adding an Alt-Svc header field to responses.

```
Alt-Svc     = 1#( alternate *( OWS ";" OWS parameter ) )
alternate   = protocol-id "=" [ uri-host ] ":" port
protocol-id = <ALPN protocol identifier>
```

For example:

```
Alt-Svc: http2=:8000
```

This indicates that the "http2" protocol on the same host using the indicated port (in this case, 8000).

Alt-Svc can also contain a host:

```
Alt-Svc: http2-tls=other.example.com:443
```

This indicates that all resources on the origin are available using the "http2-tls" profile on other.example.com port 443.

It can also have multiple values:

```
Alt-Svc: http2=:8000, http2-tls=other.example.com:443
```

The value(s) advertised by Alt-Svc can be used by clients to open a new connection to one or more alternate services immediately, or simultaneously with subsequent requests on the same connection.

Intermediaries MUST NOT change or append Alt-Svc values.

3.1. Caching Alt-Svc Header Field Values

When an alternate service is advertised using Alt-Svc, it is considered fresh for 24 hours from generation of the message. This can be modified with the 'ma' (max-age') parameter;

```
Alt-Svc: http2-tls=:443;ma=3600
```

which indicates the number of seconds since the response was generated the alternate service is considered fresh for.

ma = delta-seconds

See [I-D.ietf-httpbis-p6-cache] Section 4.2.3 for details of determining response age. For example, a response:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: 600
Age: 30
Alt-Svc: http2=:8000; ma=60
```

indicates that an alternate service is available and usable for the next 60 seconds. However, the response has already been cached for 30 seconds (as per the Age header field value), so therefore the alternate service is only fresh for the 30 seconds from when this response was received, minus estimated transit time.

When an Alt-Svc response header is received from an origin, its value invalidates and replaces all cached alternate services for that origin. This includes the empty Alt-Svc header, which clears all cached alternate services for an origin.

See Section 2.1.2 for general requirements on caching alternate services.

Note that the freshness lifetime for HTTP caching (here, 600 seconds) does not affect caching of Alt-Svc values.

3.2. Indicating Alt-Svc Header Field Priority

Finally, an explicit priority can be associated with an Alt-Svc header field value by using the "pr" parameter:

```
Alt-Svc: http2-tls:8000 ;pr=64
```

See Section 2.1.3 for details of the priority mechanism.

pr = 1*2DIGIT

If the "pr" parameter is not present or is invalid, the default priority for alternate services discovered with the Alt-Svc header field is 48.

4. Security Considerations

4.1. Changing Ports

Using an alternate service implies accessing an origin's resources on an alternate port, at a minimum. An attacker that can inject alternate services and listen at the advertised port is therefore able to hijack an origin.

For example, an attacker that can add HTTP response header fields can redirect traffic to a different port on the same host using the Alt-Svc header field; if that port is under the attacker's control, they can thus masquerade as the HTTP server.

This risk can be mitigated by restricting the ability to set the Alt-Svc response header field on the origin, and restricting who can open a port for listening on that host.

4.2. Changing Hosts

When the host is changed due to the use of an alternate service, it presents an opportunity for attackers to hijack communication to an origin.

For example, if an attacker can convince a user agent to send all traffic for "innocent.example.org" to "evil.example.com" by successfully associating it as an alternate service, they can masquerade as that origin. This can be done locally (see mitigations above) or remotely (e.g., by an intermediary as a man-in-the-middle attack).

This is the reason for the requirement in Section 2.1.1 that any alternate service with a host different to the origin's be strongly authenticated with the origin's identity; i.e., presenting a certificate for the origin proves that the alternate service is authorized to serve traffic for the origin.

However, this authorization is only as strong as the method used to authenticate the alternate service. In particular, there are well-known exploits to make an attacker's certificate appear as legitimate.

Alternate services could be used to persist such an attack; for example, an intermediary could man-in-the-middle TLS-protected communication to a target, and then direct all traffic to an alternate service with a large freshness lifetime, so that the user agent still directs traffic to the attacker even when not using the intermediary.

As a result, there is a requirement in Section 2.1.2 to examine cached alternate services when a network change is detected.

4.3. Changing Protocols

When the ALPN protocol is changed due to the use of an alternate service, the security properties of the new connection to the origin can be different from that of the "normal" connection to the origin, because the protocol identifier itself implies this.

For example, if a "https://" URI had a protocol advertised that does not use some form of end-to-end encryption (most likely, TLS), it violates the expectations for security that the URI scheme implies.

Therefore, clients cannot blindly use alternate services, but instead evaluate the option(s) presented to assure that security requirements and expectations (of specifications, implementations and end users) are met.

5. References

5.1. Normative References

[I-D.ietf-httpbis-p1-messaging]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", draft-ietf-httpbis-p1-messaging-25 (work in progress), November 2013.

[I-D.ietf-httpbis-p6-cache]
Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", draft-ietf-httpbis-p6-cache-25 (work in progress), November 2013.

[I-D.ietf-tls-applayerprotoneg]
Friedl, S., Popov, A., Langley, A., and S. Emile, "Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension", draft-ietf-tls-applayerprotoneg-03 (work in progress), October 2013.

[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.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

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

5.2. Informative References

[I-D.ietf-httpbis-http2]
Belshe, M., Peon, R., Thomson, M., and A. Melnikov, "Hypertext Transfer Protocol version 2.0", draft-ietf-httpbis-http2-08 (work in progress), November 2013.

[I-D.nottingham-http2-encryption]
Nottingham, M., "Opportunistic Encryption for HTTP URIs", draft-nottingham-http2-encryption-01 (work in progress), October 2013.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.

Appendix A. Acknowledgements

Thanks to Eliot Lear, Stephen Farrell, Guy Podjarny, Stephen Ludin, Erik Nygren, Paul Hoffman, Adam Langley and Will Chan for their feedback and suggestions.

The Alt-Svc header field was influenced by the design of the Alternate-Protocol header in SPDY.

Appendix B. TODO

- GOAWAY: A GOAWAY-like frame (or just a GOAWAY modification) that allows an alternate service to be switched to might be suggested in a future revision.
- DNS: Alternate services are also amenable to DNS-based discovery. If there is sufficient interest, a future revision may include a proposal for that.

- Indicating Chosen Service: It's likely necessary for the server to know which protocol the user agent has chosen, and perhaps even the hostname (for load balancing). This could be conveyed as part of the "magic", or as a request header.
- IPV6: The intersection between Alternate Services and Happy Eyeballs [RFC6555] should be investigated.
- ALPN strings: all of the ALPN strings in this document are fictional; they need to be updated based upon that specification's progress (and the registry, eventually).
- Advice for setting headers: guidelines for servers that use the Alt-Svc header field.

Authors' Addresses

Mark Nottingham
Akamai

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

Patrick McManus
Mozilla

Email: mcmanus@ducksong.com
URI: https://mozillians.org/u/pmcmanus/