Security Considerations for Optimistic Use of HTTP Upgrade

Ben Schwartz, Meta Platforms, Inc.

HTTPBIS @ IETF 120

Document Status

- Recently adopted
- No major changes yet from pre-adoption text
- Proposed changes based on feedback from the group
  - Including some recent feedback!
  - Deprecate the "HTTP/*.*" upgrade token (#2818, slides 3-5)
  - Extend the draft to cover HTTP CONNECT (#2821, slide 6)
  - Recommend GET for future Upgrade Tokens (#2827, slide 7)
  - Note that optimistic upgrade is safe for "TLS" (#2828, slide 8)

HTTP Upgrade Tokens

Registration Procedure(s)

First Come First Served

Reference

[RFC9110, Section 16.7]

Note

Please see [RFC9110], Section 16.7 for details regarding token registrations.

Available Formats

csv

Value	Description	Expected Version Tokens	Reference
HTTP	Hypertext Transfer Protocol	any DIGIT.DIGIT (e.g., "2.0")	[RFC9110, Section 2.5]
TLS	Transport Layer Security	ANY DIGIT.DIGIT (e.g., "1.2")	[RFC2817]
WebSocket	The Web Socket Protocol		[RFC6455]
websocket	The Web Socket Protocol		[RFC6455][RFC8441]
h2c	Hypertext Transfer Protocol version 2 (HTTP/2) (OBSOLETE)		[RFC9113, Section 3.1]
connect-udp	Proxying of UDP Payloads		[RFC9298]
connect-ip	Proxying of IP Payloads		[RFC9484]

```
GET /news.html HTTP/1.1
Host: www.example.re
Connection: upgrade
Upgrade: HTTP/2, HTTP/3
```

Response

```
HTTP/1.1 101 Switching Protocols
Upgrade: HTTP/2
Connection: upgrade
```

Takeaway

HTTP/1.1 includes a protocol upgrade feature that allows the client to change protocols without having to terminate the current HTTP Connection and create a new one. Using the Upgrade and Connection HTTP headers, in combination with HTTP response status codes 101 Switching Protocols and 426 Upgrade Required, protocols can sometimes be upgraded for different or more secure operations.

Last updated: August 2, 2023

Deprecate the "HTTP/*.*" upgrade token (#2818)

[RFC9110] … defines the "HTTP/*.*" family of Upgrade Tokens… . *In HTTP/1.1, the only potentially applicable versions of this token are "0.9", "1.0", "1.1", and "2.0". However, versions 0.9, 1.0 and 1.1 would not represent an "upgrade" from HTTP/1.1, so they are not relevant. The "HTTP/2.0" upgrade token was never adopted for HTTP/2. The token "h2c" was selected instead [RFC7540], and subsequently deprecated [RFC9113]. As a result, there are no known or anticipated use cases for the "HTTP/*.*" family of upgrade tokens. Accordingly, [the IANA considerations section] deprecates the use of these tokens.*

New IANA text: *Hypertext Transfer Protocol (OBSOLETE TOKEN)*

Originally deprecated TLS/*.* as well, but this appears to be in use.

Extend the draft to cover HTTP CONNECT (#2821)

New title: *Security Considerations for Optimistic Protocol Transitions in HTTP/1.1*

New normative text:

Clients that send HTTP CONNECT requests on behalf of untrusted TCP clients MUST wait for a 2xx (Successful) response before sending any TCP payload data.

To mitigate vulnerabilities from clients that do not conform to this requirement, proxy servers MAY close the underlying connection when rejecting an HTTP CONNECT request, without processing any further data sent to the proxy server on that connection. Note that this behavior may impair performance, especially when returning a "407 (Proxy Authentication Required)" response.

Recommend GET for future Upgrade Tokens (#2827)

The "HTTP" and "TLS" Upgrade Tokens can be used with any ordinary HTTP Method. The upgraded protocol continues to provide HTTP semantics, and will convey the response to this HTTP request.

The other Upgrade Tokens presently defined do not preserve HTTP semantics, so the method is not relevant. All of these Upgrade Tokens are specified only for use with the "GET" method.

Future specifications for Upgrade Tokens SHOULD restrict their use to GET requests if the HTTP method is otherwise irrelevant. This simplifies server implementation and reduces the risk of errors when processing request bodies.

Note that optimistic upgrade is safe for "TLS" (#2828)

The "TLS" family of upgrade tokens was defined in [RFC2817], which correctly highlights the possibility of the server rejecting the upgrade. *If a client ignores this possibility and sends TLS data optimistically, confusion between TLS and HTTP/1.1 is still prevented: the first octet of a TLS connection must be 22 (ContentType.handshake), but this is not an allowed character in an HTTP/1.1 method.*

0\r\n
\r\n