Internet Engineering Task Force (IETF)
Request for Comments: 7639
Category: Standards Track
ISSN: 2070-1721

A. Hutton
Unify
J. Uberti
Google
M. Thomson
Mozilla

August 2015

The ALPN HTTP Header Field

Abstract

This specification allows HTTP CONNECT requests to indicate what protocol is intended to be used within the tunnel once established, using the ALPN header field.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7639.

Copyright Notice

Copyright (c) 2015 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

The HTTP CONNECT method (Section 4.3.6 of [RFC7231]) requests that the recipient establish a tunnel to the identified origin server and thereafter forward packets, in both directions, until the tunnel is closed. Such tunnels are commonly used to create end-to-end virtual connections through one or more proxies.

The ALPN HTTP header field identifies the protocol or protocols that the client intends to use within a tunnel that is established using CONNECT. This uses the Application-Layer Protocol Negotiation (ALPN) identifier [RFC7301].

For a tunnel that is then secured using Transport Layer Security (TLS) [RFC5246], the header field carries the same application protocol label as will be carried within the TLS handshake [RFC7301]. If there are multiple possible application protocols, all of those application protocols are indicated.

The ALPN header field carries an indication of client intent only. An ALPN identifier is used here only to identify the application protocol or suite of protocols that the client intends to use in the tunnel. No negotiation takes place using this header field. In TLS, the final choice of application protocol is made by the server from the set of choices presented by the client. Other substrates could negotiate the application protocol differently.

Proxies do not implement the tunneled protocol, though they might choose to make policy decisions based on the value of the header field. For example, a proxy could use the application protocol to select appropriate traffic prioritization.

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. The ALPN HTTP Header Field

Clients include the ALPN header field in an HTTP CONNECT request to indicate the application-layer protocol that a client intends to use within the tunnel, or a set of protocols that might be used within the tunnel.

2.1. Header Field Values

Valid values for the protocol field are taken from the "Application-Layer Protocol Negotiation (ALPN) Protocol ID" registry [ALPN-IDS] established by [RFC7301].

2.2. Syntax

The ABNF (Augmented Backus-Naur Form) syntax for the ALPN header field value is given below. It uses the syntax defined in Section 1.2 of [RFC7230].

```
ALPN            = 1#protocol-id
protocol-id     = token ; percent-encoded ALPN protocol identifier
```

ALPN protocol names are octet sequences with no additional constraints on format. Octets not allowed in tokens ([RFC7230], Section 3.2.6) MUST be percent-encoded as per Section 2.1 of [RFC3986]. Consequently, the octet representing the percent character "%" (hex 25) MUST be percent-encoded as well.

In order to have precisely one way to represent any ALPN protocol name, the following additional constraints apply:

- Octets in the ALPN protocol MUST NOT be percent-encoded if they are valid token characters except "%".
- When using percent-encoding, uppercase hex digits MUST be used.

With these constraints, recipients can apply simple string comparison to match protocol identifiers.

For example:

```
CONNECT www.example.com HTTP/1.1
Host: www.example.com
ALPN: h2, http%2F1.1
```

2.3. Usage

When used in the ALPN header field, an ALPN identifier is used to identify an entire application protocol stack, not a single protocol layer or component.

For a CONNECT tunnel that conveys a protocol secured with TLS, the value of the ALPN header field contains the same list of ALPN identifiers that will be sent in the TLS ClientHello message [RFC7301].

Where no protocol negotiation is expected to occur, such as in protocols that do not use TLS, the ALPN header field contains a single ALPN protocol identifier corresponding to the application protocol that is intended to be used. If an alternative form of protocol negotiation is possible, the ALPN header field contains the set of protocols that might be negotiated.

A proxy can use the value of the ALPN header field to more cleanly and efficiently reject requests for a CONNECT tunnel. Exposing protocol information at the HTTP layer allows a proxy to deny requests earlier, with better error reporting (such as a 403 status code). The ALPN header field can be falsified and therefore is not a sufficient basis for authorizing a request.

A proxy could attempt to inspect packets to determine the protocol in use. This requires that the proxy understand each ALPN identifier. Protocols like TLS could hide negotiated protocols, or protocol negotiation details could change over time. Proxies SHOULD NOT break a CONNECT tunnel solely on the basis of a failure to recognize the protocol.

A proxy can use the ALPN header field value to change how it manages or prioritizes connections.

3. IANA Considerations

HTTP header fields are registered within the "Permanent Message Header Field Names" registry maintained by IANA [MSG-HDRS]. This document defines and registers the ALPN header field, according to [RFC3864] as follows:

Header Field Name: ALPN

Protocol: http

Status: Standard

Reference: Section 2 of this document (RFC 7639)

Change Controller: IETF (iesg@ietf.org) - Internet Engineering Task Force

4. Security Considerations

In case of using HTTP CONNECT to a TURN (Traversal Using Relays around NAT, [RFC5766]) server, the security considerations of Section 4.3.6 of [RFC7231] apply. It states that there "are significant risks in establishing a tunnel to arbitrary servers, particularly when the destination is a well-known or reserved TCP port that is not intended for Web traffic. ... Proxies that support CONNECT SHOULD restrict its use to a limited set of known ports or a configurable whitelist of safe request targets."

The ALPN header field described in this document is OPTIONAL. Clients and HTTP proxies could choose not to support it and therefore either fail to provide it or ignore it when present. If the header field is not available or is ignored, a proxy cannot identify the purpose of the tunnel and use this as input to any authorization decision regarding the tunnel. This is indistinguishable from the case where either client or proxy does not support the ALPN header field.

There is no confidentiality protection for the ALPN header field. ALPN identifiers that might expose confidential or sensitive information SHOULD NOT be sent, as described in Section 5 of [RFC7301].

The value of the ALPN header field could be falsified by a client. If the data being sent through the tunnel is encrypted (for example, with TLS [RFC5246]), then the proxy might not be able to directly inspect the data to verify that the claimed protocol is the one which is actually being used, though a proxy might be able to perform traffic analysis [TRAFFIC]. Therefore, a proxy cannot rely on the value of the ALPN header field as a policy input in all cases.

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>.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004, <http://www.rfc-editor.org/info/rfc3864>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

[RFC7230] Fielding, R. and J. Reschke, "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>.

[RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <http://www.rfc-editor.org/info/rfc7301>.

5.2. Informative References

[ALPN-IDS] IANA, "Application-Layer Protocol Negotiation (ALPN) Protocol ID", <http://www.iana.org/assignments/tls-extensiontype-values>.

[MSG-HDRS] IANA, "Permanent Message Header Field Names>", <https://www.iana.org/assignments/message-headers>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <http://www.rfc-editor.org/info/rfc5766>.

[TRAFFIC] Pironti, A., Strub, P-Y., and K. Bhargavan, "Identifying Website Users by TLS Traffic Analysis: New Attacks and Effective Countermeasures, Revision 1", 2012, <https://alfredo.pironti.eu/research/publications/full/identifying-website-users-tls-traffic-analysis-new-attacks-and-effective-counterme>.

Authors' Addresses

Andrew Hutton
Unify
Technology Drive
Nottingham NG9 1LA
United Kingdom

Email: andrew.hutton@unify.com

Justin Uberti
Google
747 6th Street South
Kirkland, WA 98033
United States

Email: justin@uberti.name

Martin Thomson
Mozilla
331 East Evelyn Avenue
Mountain View, CA 94041
United States

Email: martin.thomson@gmail.com