Internet Engineering Task Force	A. Hutton
Internet-Draft	Unify
Intended status: Standards Track	J. Uberti
Expires: February 19, 2015	Google
	M. Thomson
	Mozilla
	August 18, 2014

The Tunnel-Protocol HTTP Request Header Field

draft-ietf-httpbis-tunnel-protocol-00

Abstract

This specification allows HTTP CONNECT requests to indicate what protocol will be used within the tunnel once established, using the Tunnel-Protocol request header field.

Status of This Memo

Copyright Notice

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, which may then be secured using TLS (Transport Layer Security, [RFC5246]).

The HTTP Tunnel-Protocol header field identifies the protocol that will be spoken within the tunnel, using the application layer next protocol identifier [RFC7301] specified for TLS [RFC5246]".

When CONNECT is used to establish a TLS tunnel, the Tunnel-Protocol header field may be used to carry the same next protocol label as was carried within the TLS handshake. However, the HTTP-Protocol is an indication rather a negotiation since HTTP proxies do not implement the tunneled protocol.

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 Tunnel-Protocol HTTP Request Header Field

Clients include the Tunnel-Protocol Request Header field in a HTTP Connect request to indicate the application layer protocol used within the tunnel.

2.1. Header Field Values

Valid values for the protocol field are taken from the registry established in [RFC7301].

2.2. Syntax

The ABNF (Augmented Backus-Naur Form) syntax for the Tunnel-Protocol header field is given below. It is based on the Generic Grammar defined in Section 2 of [RFC7230].

Tunnel-Protocol = "Tunnel-Protocol":" 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 "%", and
- 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 turn_server.example.com:5349 HTTP/1.1
Host: turn_server.example.com:5349
Tunnel-Protocol: turn
```

3. IANA Considerations

To Be Added

4. Security Considerations

In case of using HTTP CONNECT to a TURN server the security consideration of [RFC7231], Section-4.3.6] 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 Tunnel-Protocol request header field described in this document is an optional header and HTTP Proxies may of course not support the header and therefore ignore it. If the header is not present or ignored then the proxy has no explicit indication as to the purpose of the tunnel on which to provide consent, this is the generic case that exists without the Tunnel-Protocol header.

5. References

5.1. Normative References

[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.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

[RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

[RFC7301] Friedl, S., Popov, A., Langley, A. and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, July 2014.

5.2. Informative References

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

Authors' Addresses

Andrew Hutton
Unify
Technology Drive
Nottingham, NG9 1LA
UK
EMail: andrew.hutton@unify.com

Justin Uberti
Google
747 6th Ave S
Kirkland, WA 98033
US
EMail: justin@uberti.name

Martin Thomson
Mozilla
331 E Evelyn Street
Mountain View, CA 94041
US
EMail: martin.thomson@gmail.com