               HTTP Connect - Tunnel Protocol For WebRTC
                draft-hutton-httpbis-connect-protocol-00

Abstract

   This document describes a mechanism to enable HTTP Clients to provide
   an indication within a HTTP Connect request as to which protocol will
   be used within the tunnel established to the Server identified by the
   target resource. The tunneled protocol is declared using the Tunnel-
   Protocol HTTP Request header field. Label usage relating to the use
   of HTTP Connect by WebRTC clients (e.g. turn, webrtc) are described
   in this document.

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 destination origin server
   identified by the request-target 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 RTCWEB use cases and requirements document
   [I-D.ietf-rtcweb-use-cases-and-requirements] includes a requirement
   that a WebRTC Client must be able to send streams and data to a peer
   in the presence of Firewalls that only allow traffic via a HTTP
   Proxy, when Firewall policy allows WebRTC traffic. To facilitate
   this and to allow such a HTTP Proxy to be provided with an indication
   that WebRTC related real-time media is to be included in the tunnel
   this specification defines the Tunnel-Protocol Request header field
   and associated labels. This allows the proxy to identify the
   protocol being used in the tunnel as early as possible therefore
   enabling the proxy to make informed policy decisions. The type of
   policy decisions the proxy may make is not specified here but may
   include rejecting the request with a HTTP status code responses or
   prioritizing connections. As described in Section 4.3.6 of [RFC7231]

and 2xx response indicates consent for the client to switch to tunnel
mode.

The HTTP Tunnel-Protocol header field may be used in conjunction with
and complements the application layer next protocol extension
[I-D.ietf-tls-applayerprotoneg] specified for TLS [RFC5246]". In the
scenario where the HTTP Connect is used to establish a TLS tunnel
then the HTTP Tunnel-Protocol may be used to carry the same next
protocol label as carried within the TLS handshake. However, the
Tunnel-Protocol is an indication rather a negotiation since the HTTP
Proxy does not implement the tunneled protocol. ALPN Labels are
already defined for TURN in [I-D.patil-tram-alpn] and WebRTC
[I-D.thomson-rtcweb-alpn] and are re-used here.

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. Use Cases

The following two use cases are considered:

o The WebRTC Client issues a HTTP CONNECT request to the HTTP proxy
   with the TURN server address in the Request URI.

o The WebRTC Client issues a HTTP CONNECT request to the HTTP proxy
   with the TCP address of a WebRTC peer in the Request URI. This is
   used in the case of establishing ICE-TCP [RFC6544] with a WebRTC
   Peer.

3. The Tunnel-Protocol HTTP Request Header Field

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

3.1. Header Field Values

Valid values for the protocol field are taken from the registry
established in [I-D.ietf-tls-applayerprotoneg]. For the purposes of
WebRTC, the values "webrtc" [I-D.thomson-rtcweb-alpn] and "turn"
[I-D.patil-tram-alpn] are applicable.

3.2. 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 | protocol-extension

   protocol = "webrtc" | "turn"

   protocol-extension = token

3.3. TURN as the Tunnel Protocol

   The RTCWEB transports specification [I-D.ietf-rtcweb-transports]
   requires that a WebRTC client support the modes of TURN that uses TCP
   and TLS between the client and the TURN server in order to deal with
   firewalls blocking UDP traffic. In the case where HTTP Connect is
   used to establish a tunnel to the TURN server the client SHOULD
   include the "Tunnel-Protocol" header field with the value "turn"
   [I-D.patil-tram-alpn] as shown in the example below.

           CONNECT turn_server.example.com:5349 HTTP/1.1
           Host: turn_server.example.com:5349
           Tunnel-Protocol: turn

3.4. ICE-TCP / WebRTC as the Tunnel Protocol

   [I-D.ietf-rtcweb-transports] also requires that a WebRTC client
   support ICE-TCP [RFC6544] as a mechanism to allow webrtc applications
   to communicate to peers with public IP addresses across UDP-blocking
   firewalls without using a TURN server. In this case the client
   SHOULD include the "Tunnel-Protocol" header field with the value
   "webrtc" [I-D.thomson-rtcweb-alpn] as shown in the example below.

           CONNECT 198.51.100.0:8999 HTTP/1.1
           Host: 198.51.100.0:8999
           Tunnel-Protocol: webrtc

   Note: The protocol "c_webrtc" described in [I-D.thomson-rtcweb-alpn]
   is not relevent in this context and when used at the TLS layer the
   client SHOULD use "webrtc" in the Tunnel-Protocol header. OPEN ISSUE
   - Is this correct?

4. IANA Considerations

    To Be Added

5. 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.

6. References

6.1. Normative References

    [I-D.patil-tram-alpn]
              Patil, P., Reddy, T., Salgueiro, G., and M. Petit-
              Huguenin, "Application Layer Protocol Negotiation (ALPN)
              for Session Traversal Utilities for NAT (STUN)", draft-
              patil-tram-alpn-00 (work in progress), April 2014.

    [I-D.thomson-rtcweb-alpn]
              Thomson, M., "Application Layer Protocol Negotiation for
              Web Real-Time Communications (WebRTC)", draft-thomson-
              rtcweb-alpn-00 (work in progress), April 2014.

    [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

    [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.

6.2. Informative References

   [I-D.ietf-rtcweb-transports]
              Alvestrand, H., "Transports for RTCWEB", draft-ietf-
              rtcweb-transports-05 (work in progress), June 2014.

   [I-D.ietf-rtcweb-use-cases-and-requirements]
              Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-
              Time Communication Use-cases and Requirements", draft-
              ietf-rtcweb-use-cases-and-requirements-14 (work in
              progress), February 2014.

   [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-05
              (work in progress), March 2014.

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

   [RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach,
              "TCP Candidates with Interactive Connectivity
              Establishment (ICE)", RFC 6544, March 2012.

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