   **Application Layer Protocol Negotiation (ALPN) for Session Traversal
                      Utilities for NAT (STUN)
                     draft-ietf-tram-alpn-00**

Abstract

   An Application Layer Protocol Negotiation (ALPN) label for the
   Session Traversal Utilities for NAT (STUN) protocol is defined in
   this document to allow the application layer to negotiate STUN within
   the Transport Layer Security (TLS) connection. The STUN ALPN
   protocol identifier applies to both TLS and Datagram Transport Layer
   Security (DTLS).

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 January 24, 2015.

Copyright Notice

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

STUN can be securely transported using TLS-over-TCP (referred to as
TLS [RFC5246]), as specified in [RFC5389], or TLS-over-UDP (referred
to as DTLS [RFC6347]), as specified in
[I-D.petithuguenin-tram-turn-dtls].

ALPN [I-D.ietf-tls-applayerprotoneg] enables an endpoint to
positively identify STUN protocol uses in TLS/DTLS and distinguish
them from other TLS/DTLS protocols. With ALPN, the client sends the
list of supported application protocols as part of the TLS/DTLS
ClientHello message. The server chooses a protocol and sends the
selected protocol as part of the TLS/DTLS ServerHello message. The
application protocol negotiation can thus be accomplished within the
TLS/DTLS handshake, without adding network round-trips, and allows
the server to associate a different certificate with each application
protocol, if desired.

For example, a firewall could block all outgoing traffic except for
TCP traffic to specific ports (e.g., 443 for HTTPS). A TURN server
listening on its default ports (3478 for TCP/UDP, 5349 for TLS) would
not be reachable in this case. However, despite the restrictions
imposed by the firewall, the TURN server can still be reached on the
allowed HTTPS port if an ALPN STUN protocol identifier is used to
establish the STUN application layer protocol as part of the TLS
handshake. In this case, the STUN ALPN identifier sent by the client
will be used by the server to identify that the client intends to
make a TURN request and it must act as a TURN server to relay the
traffic to and from the remote peer. Similarly, with Quick UDP
Internet Connections (QUIC) [QUIC], a UDP-based transport protocol

that operates under SPDY [I-D.mbelshe-httpbis-spdy], a TURN server
could be operated on the same ports as that of a SPDY server.

This document defines an entry ("stun") in the "Application Layer
Protocol Negotiation (ALPN) Protocol IDs" registry established by
[I-D.ietf-tls-applayerprotoneg] to identify the STUN protocol.

[[TODO: In various offline discussions some have expressed a desire
to add an additional ALPN protocol identifier for TURN (see IANA
Considerations below for example registration). ALPN can be used
more granularly to externally identify more of the protocol variants
and their different properties (i.e., STUN and TURN over TLS/DTLS).
The advantage in dividing it this way is that these different forms
can be externally identified (obviously, there isn't any inherent
value in the different identifiers from within the TLS
handshake).There are two main disadvantages. the first is that this
two application protocol approach may make implementations more
complicated/confusing. The second is that there may be difficulty in
differentiating the two with ALPN when TURN was specifically designed
to be able to run on the same port as STUN usage (in section 13 of
RFC 5389). Section 4.1.1.2 of RFC 5245 explicitly says that "If the
Allocate request is rejected because the server lacks resources to
fulfill it, the agent SHOULD instead send a Binding request to obtain
a server reflexive candidate." Does that prove there is no need to
differentiate TURN and STUN request on UDP/TCP or TLS and now DTLS?
Are there sufficiently meaningful differences between the usages to
warrant separate STUN and TURN ALPN identifiers?]]

2. Terminology

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].

3. IANA Considerations

The following entry is to be added to the "Application Layer Protocol
Negotiation (ALPN) Protocol IDs" registry established by
[I-D.ietf-tls-applayerprotoneg].

The "stun" label identifies STUN over TLS/DTLS:

   Protocol: Session Traversal Utilities for NAT (STUN)

   Identification Sequence: 0x73 0x74 0x75 0x6E ("stun")

   Specification: This document (RFCXXXX)

[[TODO: Shown only as an example. Remove the below registry entry if
open issue above dictates a single STUN ALPN identifier is
sufficient.]]

The "turn" label identifies TURN over TLS/DTLS:

   Protocol: Traversal Using Relays around NAT (TURN)

   Identification Sequence: 0x74 0x75 0x72 0x6E ("turn")

   Specification: This document (RFCXXXX)

[4](#). Security Considerations

The ALPN STUN protocol identifier does not introduce any specific
security considerations beyond those detailed in the TLS ALPN
Extension specification [I-D.ietf-tls-applayerprotoneg]. It also
does not impact the security of TLS/DTLS session establishment nor
the application data exchange.

[5](#). Acknowledgements

This work benefited from the discussions and invaluable input by the
various members of the TRAM working group. These include Simon
Perrault, Paul Kyzivat, and Andrew Hutton. Special thanks to Martin
Thomson and Oleg Moskalenko for their constructive comments,
suggestions, and early reviews that were critical to the formulation
and refinement of this document.

[6](#). References

[6.1](#). Normative References

[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.

[I-D.mbelshe-httpbis-spdy]
           Belshe, M. and R. Peon, "SPDY Protocol", draft-mbelshe-
           httpbis-spdy-00 (work in progress), February 2012.

[I-D.petithuguenin-tram-turn-dtls]
           Petit-Huguenin, M. and G. Salgueiro, "Datagram Transport
           Layer Security (DTLS) as Transport for Traversal Using
           Relays around NAT (TURN)", draft-petithuguenin-tram-turn-
           dtls-00 (work in progress), January 2014.

    [QUIC]   http://www.ietf.org/proceedings/88/slides/
                slides-88-tsvarea-10.pdf, "QUIC Slide Deck at IETF88,", .

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

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

    [RFC5389]   Rosenberg, J., Mahy, R., Matthews, P., and D. Wing,
                "Session Traversal Utilities for NAT (STUN)", RFC 5389,
                October 2008.

    [RFC6347]   Rescorla, E. and N. Modadugu, "Datagram Transport Layer
                Security Version 1.2", RFC 6347, January 2012.

6.2. Informative References

    [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, April 2010.

Authors' Addresses

    Prashanth Patil
    Cisco Systems, Inc.
    Bangalore
    India

    Email: praspati@cisco.com

    Tirumaleswar Reddy
    Cisco Systems, Inc.
    Cessna Business Park, Varthur Hobli
    Sarjapur Marathalli Outer Ring Road
    Bangalore, Karnataka 560103
    India

    Email: tireddy@cisco.com

      Gonzalo Salgueiro
      Cisco Systems, Inc.
      7200-12 Kit Creek Road
      Research Triangle Park, NC 27709
      US

      Email: gsalguei@cisco.com

      Marc Petit-Huguenin
      Jive Communications
      1275 West 1600 North, Suite 100
      Orem, UT 84057
      USA

      Email: marcph@getjive.com