Internet Engineering Task Force (IETF)
Request for Comments: 7350
Updates: 5389, 5928
Category: Standards Track
ISSN: 2070-1721
August 2014

M. Petit-Huguenin
Jive Communications
G. Salgueiro
Cisco Systems

Datagram Transport Layer Security (DTLS) as Transport for Session Traversal Utilities for NAT (STUN)

Abstract

This document specifies the usage of Datagram Transport Layer Security (DTLS) as a transport protocol for Session Traversal Utilities for NAT (STUN). It provides guidance on when and how to use DTLS with the currently standardized STUN usages. It also specifies modifications to the STUN and Traversal Using Relay NAT (TURN) URIs and to the TURN resolution mechanism to facilitate the resolution of STUN and TURN URIs into the IP address and port of STUN and TURN servers supporting DTLS as a transport protocol. This document updates RFCs 5389 and 5928.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

STUN [RFC5389] defines Transport Layer Security (TLS) over TCP (simply referred to as TLS [RFC5246]) as the transport for STUN due to additional security advantages it offers over plain UDP or TCP transport. But, TCP (and thus TLS-over-TCP) is not an optimal transport when STUN is used for its originally intended purpose, which is to support multimedia sessions. This is a well documented and understood transport limitation for real-time communications.

DTLS-over-UDP (referred to in this document as simply DTLS [RFC6347]) offers the same security advantages as TLS-over-TCP, but without the undesirable concerns.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "must" or "Must"), they have their usual English meanings, and are not to be interpreted as RFC 2119 key words.

3. DTLS as Transport for STUN

STUN [RFC5389] defines three transports: UDP, TCP, and TLS. This document adds DTLS as a valid transport for STUN.

STUN over DTLS MUST use the same retransmission rules as STUN over UDP (as described in Section 7.2.1 of [RFC5389]). It MUST also use the same rules that are described in Section 7.2.2 of [RFC5389] to verify the server identity. Instead of TLS_RSA_WITH_AES_128_CBC_SHA, which is the default cipher suite for STUN over TLS, implementations of STUN over DTLS, and deployed clients and servers, MUST support TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 and TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, and MAY support other cipher suites. Perfect Forward Secrecy (PFS) cipher suites MUST be preferred over non-PFS cipher suites. Cipher suites with known weaknesses, such as those based on (single) DES and RC4, MUST NOT be used. Implementations MUST disable TLS-level compression. The same rules established in Section 7.2.2 of [RFC5389] for keeping open and closing TCP/TLS connections MUST be used as well for DTLS associations.

In addition to the path MTU rules described in Section 7.1 of [RFC5389], if the path MTU is unknown, the actual STUN message needs to be adjusted to take into account the size of the (13-byte) DTLS Record header, the MAC size, and the padding size.

By default, STUN over DTLS MUST use port 5349, the same port number as STUN over TLS. However, the Service Record (SRV) procedures can be implemented to use a different port (as described in Section 9 of [RFC5389]). When using SRV records, the service name MUST be set to "stuns" and the protocol name to "udp".

Classic STUN [RFC3489] (which was obsoleted by [RFC5389]) defines only UDP as a transport, and DTLS MUST NOT be used. Any STUN request or indication without the magic cookie (see Section 6 of [RFC5389]) over DTLS MUST always result in an error.

4. STUN Usages

Section 7.2 of [RFC5389] states that STUN usages must specify which transport protocol is used. The following sections discuss if and how the existing STUN usages are used with DTLS as the transport. Future STUN usages MUST take into account DTLS as a transport and discuss its applicability. In all cases, new STUN usages MUST explicitly state if implementing the denial-of-service countermeasure described in Section 4.2.1 of [RFC6347] is mandatory.

4.1. NAT Discovery Usage

As stated by Section 13 of [RFC5389], "...TLS provides minimal security benefits..." for this particular STUN usage. DTLS will also similarly offer only limited benefit. This is because the only mandatory attribute that is TLS/DTLS protected is the XOR-MAPPED-ADDRESS, which is already known by an on-path attacker, since it is the same as the source address and port of the STUN request. On the other hand, using TLS/DTLS will prevent an active attacker to inject XOR-MAPPED-ADDRESS in responses. The TLS/DTLS transport will also protect the SOFTWARE attribute, which can be used to find vulnerabilities in STUN implementations.

Regardless, this usage is rarely used by itself, since using TURN [RFC5766] with Interactive Connectivity Establishment (ICE) [RFC5245] is generally indispensable, and TURN provides the same NAT Discovery feature as part of an allocation creation. In fact, with ICE, the NAT Discovery usage is only used when there is no longer any resource available for new allocations in the TURN server.

A STUN server implementing the NAT Discovery usage and using DTLS MUST implement the denial-of-service countermeasure described in Section 4.2.1 of [RFC6347].

4.1.1. DTLS Support in STUN URIs

This document does not make any changes to the syntax of a STUN URI [RFC7064]. As indicated in Section 3.2 of [RFC7064], secure transports like STUN over TLS, and now STUN over DTLS, MUST use the "stuns" URI scheme.

The <host> value MUST be used when using the rules in Section 7.2.2 of [RFC5389] to verify the server identity. A STUN URI containing an IP address MUST be rejected, unless the domain name is provided by the same mechanism that provided the STUN URI, and that domain name can be passed to the verification code.

4.2. Connectivity Check Usage

Using DTLS would hide the USERNAME, PRIORITY, USE-CANDIDATE, ICE-CONTROLLED, and ICE-CONTROLLING attributes. But, because MESSAGE-INTEGRITY protects the entire STUN response using a password that is known only by looking at the Session Description Protocol (SDP) exchanged, it is not possible for an attacker that does not have access to this SDP to inject an incorrect XOR-MAPPED-ADDRESS, which would subsequently be used as a peer reflexive candidate.

Adding DTLS on top of the connectivity check would delay, and consequently impair, the ICE process. Adding additional round trips to ICE is undesirable, so much that there is a proposal ([ICE-DTLS]) to use the DTLS handshake used by the WebRTC Secure Real-time Transport Protocol (SRTP) streams as a replacement for the connectivity checks.

STUN URIs are not used with this usage.

4.3. Media Keep-Alive Usage

When STUN Binding Indications are being used for media keep-alive (described in Section 10 of [RFC5245]), it runs alongside an RTP or RTP Control Protocol (RTCP) session. It is possible to send these media keep-alive packets inside a separately negotiated non-SRTP DTLS session if DTLS-SRTP [RFC5764] is used, but that would add overhead, with minimal security benefit.

STUN URIs are not used with this usage.

4.4. SIP Keep-Alive Usage

The SIP keep-alive (described in [RFC5626]) runs inside a SIP flow. This flow would be protected if a SIP over DTLS transport mechanism is implemented (such as described in [SIP-DTLS]).

STUN URIs are not used with this usage.

4.5. NAT Behavior Discovery Usage

The NAT Behavior Discovery usage is Experimental and to date has never been effectively deployed. Despite this, using DTLS would add the same security properties as for the NAT Discovery usage (Section 4.1).

The STUN URI can be used to access the NAT Discovery feature of a NAT Behavior Discovery server, but accessing the full features would require definition of a "stun-behaviors:" URI, which is out of scope for this document.

A STUN server implementing the NAT Behavior Discovery usage and using DTLS MUST implement the denial-of-service countermeasure described in Section 4.2.1 of [RFC6347].

4.6. TURN Usage

TURN [RFC5766] defines three combinations of transports/allocations: UDP/UDP, TCP/UDP, and TLS/UDP. This document adds DTLS/UDP as a valid combination. A TURN server using DTLS MUST implement the denial-of-service countermeasure described in Section 4.2.1 of [RFC6347].

[RFC6062] states that TCP allocations cannot be obtained using a UDP association between client and server. The fact that DTLS uses UDP implies that TCP allocations MUST NOT be obtained using a DTLS association between client and server.

By default, TURN over DTLS uses port 5349, the same port number as TURN over TLS. However, the SRV procedures can be implemented to use a different port (as described in Section 6 of [RFC5766]). When using SRV records, the service name MUST be set to "turns" and the protocol name to "udp".

4.6.1. DTLS Support in TURN URIs

This document does not make any changes to the syntax of a TURN URI [RFC7065]. As indicated in Section 3 of [RFC7065], secure transports like TURN over TLS, and now TURN over DTLS, MUST use the "turns" URI scheme. When using the "turns" URI scheme to designate TURN over DTLS, the transport value of the TURN URI, if set, MUST be "udp".

The <host> value MUST be used when using the rules in Section 7.2.2 of [RFC5389] to verify the server identity. A TURN URI containing an IP address MUST be rejected, unless the domain is provided by the same mechanism that provided the TURN URI, and that domain name can be passed to the verification code.

4.6.2. Resolution Mechanism for TURN over DTLS

This document defines a new Straightforward-Naming Authority Pointer (S-NAPTR) application protocol tag: "turn.dtls".

The <transport> component, as provisioned or resulting from the parsing of a TURN URI, is passed without modification to the TURN resolution mechanism defined in Section 3 of [RFC5928], but with the following alterations to that algorithm:

- The acceptable values for the transport name are extended with the addition of "dtls".

- The acceptable values in the ordered list of supported TURN transports is extended with the addition of "Datagram Transport Layer Security (DTLS)".

- The resolution algorithm check rules list is extended with the addition of the following step:

  If <secure> is true and <transport> is defined as "udp" but the list of TURN transports supported by the application does not contain DTLS, then the resolution MUST stop with an error.

- The 5th rule of the resolution algorithm check rules list is modified to read like this:

  If <secure> is true and <transport> is not defined but the list of TURN transports supported by the application does not contain TLS or DTLS, then the resolution MUST stop with an error.

- Table 1 is modified to add the following line:

<secure>	<transport>	TURN Transport
true	"udp"	DTLS

- In step 1 of the resolution algorithm, the default port for DTLS is 5349.

- In step 4 of the resolution algorithm, the following is added to the list of conversions between the filtered list of TURN transports supported by the application and application protocol tags:

  "turn.dtls" is used if the TURN transport is DTLS.

Note that using the resolution mechanism in [RFC5928] does not imply that additional round trips to the DNS server will be needed (e.g., the TURN client will start immediately if the TURN URI contains an IP address).

5. Security Considerations

STUN over DTLS as a STUN transport does not introduce any specific security considerations beyond those for STUN over TLS detailed in [RFC5389].

The usage of "udp" as a transport parameter with the "stuns" URI scheme does not introduce any specific security issues beyond those discussed in [RFC7064].

TURN over DTLS as a TURN transport does not introduce any specific security considerations beyond those for TURN over TLS detailed in [RFC5766].

The usage of "udp" as a transport parameter with the "turns" URI scheme does not introduce any specific security issues beyond those discussed in [RFC7065].

The new S-NAPTR application protocol tag defined in this document as well as the modifications this document makes to the TURN resolution mechanism described in [RFC5928] do not introduce any additional security considerations beyond those outlined in [RFC5928].

6. IANA Considerations

6.1. S-NAPTR Application Protocol Tag

This specification contains the registration information for one S-NAPTR application protocol tag in the "Straightforward-NAPTR (S-NAPTR) Parameters" registry under "S-NAPTR Application Protocol Tags" (in accordance with [RFC3958]).

Application Protocol Tag: turn.dtls
Intended Usage: See Section 4.6.2
Interoperability considerations: N/A
Security considerations: See Section 5
Relevant publications: This document
Contact information: Marc Petit-Huguenin <petithug@acm.org>
Author/Change controller: The IESG

6.2. Service Name and Transport Protocol Port Number

This specification contains the registration information for two Service Name and Transport Protocol Port Numbers in the "Service Names and Transport Protocol Port Numbers/Service Name and Transport Protocol Port Number" registry (in accordance with [RFC6335]).

6.2.1. The "stuns" Service Name

IANA has modified the following entry in the registry "Service Names and Transport Protocol Port Numbers/Service Name and Transport Protocol Port Number":

Service Name: stuns
PortNumber: 5349
Transport Protocol: udp
Description: Reserved for a future enhancement of STUN
Assignee:
Contact:
Reference: RFC 5389

So that it contains the following:

Service Name: stuns
Port Number: 5349
Transport Protocol: udp
Description: STUN over DTLS
Assignee: IESG
Contact: IETF Chair <chair@ietf.org>
Reference: RFC 7350
Assignment Notes: This service name was initially created by RFC 5389.

6.2.2. The "turns" Service Name

IANA has modified the following entry in the registry "Service Names and Transport Protocol Port Numbers/Service Name and Transport Protocol Port Number":

Service Name: turns
Port Number: 5349
Transport Protocol: udp
Description: Reserved for a future enhancement of TURN
Assignee:
Contact:
Reference: RFC 5766

So that it contains the following:

Service Name: turns
Port Number: 5349
Transport Protocol: udp
Description: TURN over DTLS
Assignee: IESG
Contact: IETF Chair <chair@ietf.org>
Reference: RFC 7350
Assignment Notes: This service name was initially created by RFC 5766.

7. Acknowledgements

Thanks to Alan Johnston, Oleg Moskalenko, Simon Perreault, Thomas Stach, Simon Josefsson, Roni Even, Kathleen Moriarty, Benoit Claise, Martin Stiemerling, Jari Arkko, and Stephen Farrell for the comments, suggestions, and questions that helped improve this document.

8. References

8.1. Normative References

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

[RFC3489] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[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.

[RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

[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.

[RFC5928] Petit-Huguenin, M., "Traversal Using Relays around NAT (TURN) Resolution Mechanism", RFC 5928, August 2010.

[RFC6062] Perreault, S. and J. Rosenberg, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations", RFC 6062, November 2010.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.

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

[RFC7064] Nandakumar, S., Salgueiro, G., Jones, P., and M. Petit-Huguenin, "URI Scheme for the Session Traversal Utilities for NAT (STUN) Protocol", RFC 7064, November 2013.

[RFC7065] Petit-Huguenin, M., Nandakumar, S., Salgueiro, G., and P. Jones, "Traversal Using Relays around NAT (TURN) Uniform Resource Identifiers", RFC 7065, November 2013.

8.2. Informative References

[ICE-DTLS] Thomson, M., "Using Datagram Transport Layer Security (DTLS) For Interactivity Connectivity Establishment (ICE) Connectivity Checking: ICE-DTLS", Work in Progress, March 2012.

[SIP-DTLS] Jennings, C. and N. Modadugu, "Session Initiation Protocol (SIP) over Datagram Transport Layer Security (DTLS)", Work in Progress, October 2007.

Appendix A. Examples

Table 1 shows how the <secure>, <port>, and <transport> components are populated for a TURN URI that uses DTLS as its transport. For all these examples, the <host> component is populated with "example.net".

URI	<secure>	<port>	<transport>
turns:example.net?transport=udp	true		DTLS

Table 1

With the DNS Resource Records (RRs) in Figure 1 and an ordered TURN transport list of {DTLS, TLS, TCP, UDP}, the resolution algorithm will convert the TURN URI "turns:example.net" to the ordered list of IP address, port, and protocol tuples in Table 2.

```
example.net.
IN NAPTR 100 10 "" RELAY:turn.udp:turn.dtls "" datagram.example.net.
IN NAPTR 200 10 "" RELAY:turn.tcp:turn.tls "" stream.example.net.

datagram.example.net.
IN NAPTR 100 10 S RELAY:turn.udp "" _turn._udp.example.net.
IN NAPTR 200 10 S RELAY:turn.dtls "" _turns._udp.example.net.

stream.example.net.
IN NAPTR 100 10 S RELAY:turn.tcp "" _turn._tcp.example.net.
IN NAPTR 200 10 A RELAY:turn.tls "" a.example.net.

_turn._udp.example.net.
IN SRV   0 0 3478 a.example.net.

_turn._tcp.example.net.
IN SRV   0 0 5000 a.example.net.

_turns._udp.example.net.
IN SRV   0 0 5349 a.example.net.

a.example.net.
IN A     192.0.2.1
```

Figure 1

Order	Protocol	IP address	Port
1	DTLS	192.0.2.1	5349
2	TLS	192.0.2.1	5349

Table 2

Authors' Addresses

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

EMail: marcph@getjive.com

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

EMail: gsalguei@cisco.com