Operations and Management Area Working Group
Internet-Draft
Updates: RFC8907 (if approved)
Intended status: Standards Track
Expires: 21 September 2024

T. Dahm
J. Heasley
NTT
D.C. Medway Gash
Cisco Systems, Inc.
A. Ota
Google Inc.
20 March 2024

TACACS+ TLS 1.3
draft-ietf-opsawg-tacacs-tls13-06

Abstract

The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol [(RFC_8907)] provides device administration for routers, network access servers and other networked computing devices via one or more centralized servers. This document adds Transport Layer Security (TLS 1.3) support to TACACS+ and obsoletes former inferior security mechanisms.

This document updates RFC 8907.

Commenté [BMI1]: Please see the policy at RFC EDITOR GUIDELINES AND PROCEDURES (rfc-editor.org)

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP14] when, and only when, they appear in all capitals, as shown here.

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 https://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 21 September 2024.

Table of Contents

1. Introduction

The Terminal Access Controller Access-Control System Plus (TACACS+) ~~+~~ Protocol [RFC8907] provides device administration for routers, network access servers and other networked computing devices via one or more centralized servers. The protocol provides authentication, authorization and accounting services (AAA) for TACACS+ clients within the ~~Device device administration~~ Administration use case.

Commenté [BMI2]: To be consistent with RFC8907

While the content of the protocol is highly sensitive, TACACS+ lacks ~~modern and/or~~ effective confidentiality, integrity, and authentication of the connection and network traffic between the server and client, requiring secure transport to safeguard ~~the~~a deployment. The existing TACACS+ mechanisms are extremely weak as described in ~~section~~ Section 10 of ~~TACACS+ Protocol~~ [RFC8907].

To address these deficiencies, this document updates the TACACS+ protocol to use TLS 1.3 [RFC8446] authentication and encryption, and obsoletes the use of its former mechanisms (Section 10.5 of [RFC8907]).

2. Technical Definitions

The terms defined in ~~section~~ Section 3 of ~~TACACS+ Protocol~~ [RFC8907] are fully applicable here and will not be repeated. The following terms are also used in this document.

2.1. Obfuscation

Obfuscation is the inferior form of encryption used in TACACS+, labelled as such in ~~[RFC8907],~~ Section 10.5.2 of [RFC8907] to indicate that it is not encryption and is insufficient.

2.2. Non-TLS Connection

This is ~~another~~ term refers to ~~for~~ a connection defined in ~~the historic TACACS+ Protocol~~ [RFC8907]. It is a connection without TLS and therefore being plaintext or possibly using unsecure TACACS+ authentication and obfuscation. The use of well-known TCP/IP server port number 49 is specified as the default for Non-TLS connections.

Non-TLS connections SHOULD NOT be used for new TACACS+ deployments.

2.3. TLS Connection

A TLS connection is a TCP/IP connection with TLS authentication and encryption used by TACACS+ for transport. A TLS connection for TACACS+ is always between one TACACS+ client and one TACACS+ Server.

2.4. TACACS+ Server

A TACACS+ server is an instance of the server as defined in Section 3.2 of [RFC8907]~~, Section 3.2~~ that responds to TACACS+ traffic bound to ~~on~~ a specific Port number in a host. This document distinguishes the TACACS+ server instance from the host itself. A host may have multiple TACACS+ server instances ~~installed~~running, each listening to different port numbers.

2.5. Peer

The peer of a TACACS+ client (or server) in the context of a TACACS+ connection, is ~~the~~ a server (or client). Together, the ends of a TACACS+ connection are referred to as the peers.

3. TACACS+ over TLS

TACACS+ over TLS takes the protocol defined in TACACS+ Protocol [RFC8907], removes the option for MD5 obfuscation, and specifies the use of TLS (version 1.3 or later) for transport using a new well-known default server port number. The next sections provide further details and guidance.

Commenté [BMI3]: I would avoid using "encryption" here to be consistent with this part of RFC8907: "but the algorithm would not meet modern standards and so will not be termed as encryption in this document.¶"

Commenté [BMI4]: As we are not proposing to change the status of 8907 to "Historic": RFC 2026 defines a "Historic" status for documents: A specification that has been superseded by a more recent specification or is for any other reason considered to be obsolete is assigned to the "Historic" level.

Commenté [BMI5]: It is weird to have a recommendation in the definition section. Please move this one to the main document.

TLS is introduced into TACACS+ to fulfill the following requirements:

1. Confidentiality and Integrity: The MD5 obfuscation specified in ~~RFC8907~~ [RFC8907] has been shown to be insecure [RFC6151]. This prevents TACACS+ being used in a FIPS-140 compliant deployment. Securing TACACS+ protocol with TLS is intended to provide confidentiality and integrity without requiring the provision of a secured network.

2. Peer authentication: The authentication capabilities of TLS replace the pre-shared keys of obfuscation for mutual authentication.

3.1. Separating TLS Connections

All data exchanged by TACACS+ peers MUST be encrypted, including the mutual authentication of the peers. Therefore, when a TCP connection is established for the service, a TLS handshake begins immediately.

To ensure separation of TACACS+ traffic that uses TLS from that which does not (~~see~~ Section 5.3), TACACS+ over TLS will follow [RFC7605]. This entails dedicating separate TCP/IP port numbers to each protocol type.

Additionally, considering the widespread use of default settings in numerous existing TACACS+ configurations, the designated port number for TLS TACACS+ will be defined as a well-known system TCP/IP port number assigned by IANA, port [TBD] (Section 7) with the service name [TBDN] (Section 7).

Under exceptional circumstances, this document permits any other TCP port number to be configured when required by deployment specifics, but the implications in Section 5.3 ~~MUST~~ have to ~~still~~ be considered by operators.

3.2. TLS Connection

A TACACS+ client initiates a TLS connection by making a TCP connection to a configured ~~-~~server on the TACACS+ TLS ~~well-known~~ port number ([TBD]) (Section 3.1). Once the TCP connection is established, the client MUST immediately begin the TLS negotiation before sending any TACACS+ protocol data.

Implementations MUST support TLS 1.3 [RFC8446] and MAY permit TLS 1.3 session resumption. If resumption is supported, the resumption ticket_lifetime SHOULD be configurable, including a zero seconds lifetime.

Once the TLS connection is established, the exchange of TACACS+ data proceeds as normal, except that it is transmitted over TLS as TLS application data and without TACACS+ obfuscation (~~see~~ Section 4)

The connection persists until the server or client closes it. It might be closed due to an error or at the conclusion of the TACACS+ session. If Single Connection Mode has been negotiated, it might remain open after a successful session, until an error or an inactivity timeout occurs. Why it closed has no bearing on TLS resumption, unless closed by a TLS error, in which case the ticket

Commenté [BMI6]: That RFC says: "However, there is no IETF consensus on when separate ports should be used for secure and insecure variants of the same service [RFC2595] [RFC2817] [RFC6335]. The overall preference is for use of a single port, as noted in Section 6 of this document and Section 7.2 of [RFC6335], but the appropriate approach depends on the specific characteristics of the service."

Commenté [BMI7]: I don't think the use of normative language is justified here.

Commenté [BMI8]: Otherwise, we exclude the exception you described right before. Also, clients will establish connections with configured IP address/port. If not no port is configured, then the well-know will be used.

might be invalidated.

3.2.1. Cipher Requirements

Implementations MUST support the TLS 1.3 mandatory cipher suites (See TLS 1.3 [RFC8446] Section 9.1). The cipher suites offered or accepted SHOULD be configurable so that operators can adapt.

This document makes no cipher suite recommendations, but recommendations can be found in the TLS Cipher Suites section of the [RFC9325].

Commenté [BMI9]: Please cite Section 4 explicitely

3.2.2. TLS Authentication

Implementations MUST support certificate-based TLS authentication and certificate revocation bi-directionally for authentication, identity verification and policy purposes. Certificate path verification as described in Section 3.2.2.1 MUST be supported.

If the verification succeeds, the authentication is successful and the connection is permitted. Policy may impose further constraints upon the peer, allowing or denying the connection based on certificate fields or any other parameters exposed by the implementation.

Unless disabled by configuration, a peer MUST disconnect any peer that presents an invalid TLS Certificate.

Implementations MAY support TLS authentication with Pre-Shared Keys (PSKs), also known as external PSKs in TLS 1.3, which are not resumption PSKs. PSKs are considered out-of-scope for this document.

Commenté [BMI10]: What about raw public keys?

Commenté [BMI11]: Cite rfc4279

3.2.2.1. TLS Certificate Path Verification

Implementations MUST support certificate Path verification as described in [RFC5280].

Because a peer could be isolated from a remote peer's Certificate Authority (CA), implementations MUST support certificate chains (a.k.a. bundles or chains of trust), where the entire chain of the remote's certificate is stored on the local peer.

Commenté [BMI12]: What about RFC7924; should this be recommended to avoid transmitting the server's certificate and certificate chain if the client has cached that information from a previous TLS handshake?

3.3. TLS Identification

In addition to authentication of TLS certificates, implementations MUST allow operators to specify which certificate fields are to be used for peer-identification, to verify that the peer is a valid source for the received certificate and that it is permitted access to TACACS+. Implementations MUST support either:

Commenté [BMI13]: The likely mode is that certificates will be issued using a domain name, not an IP address.

So, what about RFC9525 to compare an available domain name with the certificate?

Network location based validation methods as described in [RFC5425], Section 5.2.

or

Device Identity based validation methods where the peer's identity is used in the certificate subjectName. This is applicable in deployments where the device securely supports an identity which is shared with its peer. This approach allows a peer's network location to be reconfigured without issuing a new client certificate. Only the local server mapping needs to be updated.

Implementations SHOULD support the TLS Server Name Indication extension ([RFC6066], Section 3). Policy can be applied to this attribute and it can be useful for load balancing or multiplexing at the server.

Certificate Provisioning is out of scope of this document.

Commenté [BMI14]: I wonder whether the requirement should be tweak as follows: clients SHOULD include the server domain name in the SNI extension?

Commenté [BMI15]: Not sure what is meant here.

Commenté [BMI16]: What about configuration of name/address/port number of the server?

4. Obsolescence of TACACS+ Obfuscation

~~The original draft of TACACS+~~ [RFC8907] ~~described~~ describes the obfuscation mechanism, documented in Section 5.2 of [RFC5425]~~, Section 5.2~~. ~~It is insufficient for modern purposes~~Such method is weak.

The introduction of TLS PSK, certificate peer authentication, and TLS encryption to TACACS+ replaces these former mechanisms and so obfuscation is hereby obsoleted. This section describes how the TACACS+ client and servers MUST operate with regards to the obfuscation mechanism.

Peers MUST NOT use obfuscation with TLS.

A TACACS+ client initiating a TACACS+ TLS connection MUST set the TAC_PLUS_UNENCRYPTED_FLAG bit, thereby asserting that obfuscation is not used for the session. All subsequent packets MUST have the TAC_PLUS_UNENCRYPTED_FLAG set.

A TACACS+ server that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG not set (i.e., cleared) over a TLS connection, MUST return an error of TAC_PLUS_AUTHEN_STATUS_ERROR, TAC_PLUS_AUTHOR_STATUS_ERROR, or TAC_PLUS_ACCT_STATUS_ERROR as appropriate for the TACACS+ message type, with the TAC_PLUS_UNENCRYPTED_FLAG set, and terminate the session. This behavior corresponds to that defined in ~~RFC8907~~ Section 4.5 of [RFC8907]. Data Obfuscation [RFC8907] for TAC_PLUS_UNENCRYPTED_FLAG or key mismatches.

A TACACS+ client that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG not set (i.e., cleared), MUST terminate the session, and SHOULD log this error.

5. Security Considerations

5.1. TLS

This document improves the confidentiality, integrity, and authentication of the connection and network traffic between TACACS+ peers by adding TLS support.

Simply adding TLS support to the protocol does not guarantee the protection of the server and clients. It is essential for the operators and equipment vendors to adhere to the latest best practices for ensuring the integrity of network devices and selecting secure TLS key and encryption algorithms.

[RFC9325]~~.~~ offers substantial guidance for implementing protocols that use TLS and their deployment. Those implementing and deploying Secure TACACS+ must adhere to the recommendations relevant to TLS 1.3 outlined in [RFC9325], or its subsequent versions.

This document outlines additional restrictions permissible under [RFC9325]. For example, any recommendations referring to TLS 1.2, including the mandatory support, are not relevant for Secure TACACS+ as TLS 1.3 or above is mandated.

5.1.1. TLS Use

TLS encryption SHOULD be used in deployments where both the clients and servers support it. TACACS+ servers that have TLS support MUST NOT allow N~~n~~on-TLS connections from clients ~~that do not support TLS~~, because of the threat of downgrade attacks, as described in Section 5.2. Instead, separate ~~non~~Non-TLS TACACS+ servers can be set up to cater for these clients.

Further, TLS TACACS+ servers and non-TLS TACACS+ servers SHOULD NOT be deployed on the same host. Non-TLS connections would be better served by deploying the required Non-TLS TACACS+ servers on separate hosts.

It is NOT RECOMMENDED to deploy TACACS+ without TLS authentication and encryption, including TLS using the NULL algorithm, except for within test and debug environments. Also see [RFC3365].

5.1.2. TLS 0-RTT

TLS 1.3 resumption and PSK techniques make it possible to send Early Data, a.k.a. 0-RTT data, data that is sent before the TLS handshake completes. Replay of this data is a risk. Given the sensitivity of TACACS+ data, clients MUST NOT send data until the full TLS handshake completes; that is, clients MUST NOT send 0-RTT data and servers MAY abruptly disconnect clients that do.

5.1.3. TLS Options

Implementors and operators SHOULD make use of the various RFCs to determine which TLS versions and algorithms should be supported, deprecated, obsoleted, or abandoned, in the absence of updates to this document.

Recommendations in Section 4 of [RFC9325] ~~Section 4~~, or any RFCs which obsolete it, MUST be followed.

Other useful examples are the TLS specifications themselves (TLS 1.3 [RFC8446]), which prescribes mandatory support in Section 9, and TLS Recommendations [RFC7525].

5.1.4. Unreachable Certificate Authority (CA)

Operators SHOULD be cognizant of the potential of server and/or

Commenté [BMI17]: I would delete this sentence. The normative language is not justified.

Commenté [BMI18]: Isn't the text right after sufficient? I would delete this text.

client isolation from their peer's CA by network failures. Isolation from a public key certificate's CA will cause the verification of the certificate to fail and thus TLS authentication of the peer to fail. Operators SHOULD consider loading certificate chains on devices and servers to avoid this failure.

Certificate caching and Raw Public Keys [RFC7250] are other methods to help address this, but both are out of scope for this document. Certificate fingerprints are another option.

5.1.5. TLS Server Name Indicator (SNI)

Operators SHOULD be aware that the TLS SNI extension is part of the TLS client hello, and is therefore subject to eavesdropping. Also see [RFC6066], Section 11.1.

If TLS Encrypted Client Hello becomes standardized and applicable to TLS 1.3, then it SHOULD be included in Secure TACACS+ implementation.

5.2. TACACS+ Configuration

Implementors MUST ensure that the configuration scheme introduced for enabling TLS is straightforward and leaves no room for ambiguity regarding whether TLS or ~~non~~Non-TLS will be used between the TACACS+ client and the TACACS+ server.

This document ~~introduces~~ recommends the use of a separate port number that TLS enabled TACACS+ servers will listen to. Where deployments have not overridden the defaults explicitly, TACACS+ client implementations MUST use the correct values:

* for non-TLS connection TACACS+: Port 49.

* for TLS connection TACACS+: (TBD).

Implementors MAY offer a single option for TACACS+ clients and servers to disable all ~~n~~Non-TLS TACACS+ operations. When enabled on a TACACS+ server, it will not respond to any requests from ~~n~~Non-TLS TACACS+ client connections. When enabled on a TACACS+ client, it will not establish any non-TLS TACACS+ server connections.

5.3. Well-Known TCP/IP Port

A new port is considered appropriate and superior to a "STARTTLS" command or other negotiation method because it allows:

* ease of blocking the unobfuscated or obfuscated connections by the TCP/IP port number,

* passive Intrusion Detection Systems (IDSs) monitoring the unobfuscated to be unaffected by the introduction of TLS,

* avoidance of Man in the Middle (MitM) attacks that can interfere with STARTTLS,

* and helps prevent the accidental exposure of sensitive information due to misconfiguration.

However, co-existence of inferior authentication and obfuscated,

whether an Non-TLS connection or deprecated parts that compose TLS, also presents opportunity for down-grade attacks. Causing failure of connections to the TLS-enabled service or the negotiation of shared algorithm support are two such down-grade attacks.

The simplest way to address exposure from Non-TLS connection methods is to refuse Non-TLS connections at the server entirely, perhaps using separate servers for Non-TLS connections and TLS.

Another approach is mutual configuration that requires TLS. Clients and servers SHOULD support configuration that requires peers, globally and individually, use TLS. Furthermore, peers SHOULD be configurable to limit offered or recognized TLS versions and algorithms to those recommended by standards bodies and implementers.

6. Operator Considerations

Operational and deployment considerations are spread throughout the document. While avoiding repetition, it is useful for the impatient to direct particular attention to Sections 5.2 and ~~Section~~ 5.1.5. However, it is important that the entire Section 5 is observed.

6.1. Migration

In ~~s~~section 5.2, it is mentioned that for an optimal deployment of TLS TACACS+, TLS should be universally applied throughout the deployment. However, during the migration process from a non-TLS TACACS+ deployment, operators may need to support both TLS and Non-TLS TACACS+ servers. This migration phase allows operators to gradually transition their deployments from an insecure state to a more secure one, but it is important to note that it is vulnerable to downgrade attacks. Therefore, the migration phase should be considered insecure until it is fully completed. To mitigate this hazard:

* the period where any client is configured with both TLS and ~~non~~Non-TLS servers ~~SHOULD~~ should be minimized.

* the operator ~~MUST~~ must consider the impact of mixed TLS and Non-TLS on security.

6.2. Maintaining Non-TLS TACACS+ Clients

Some TACACS+ client devices in a deployment may not implement TLS. These devices will require access to Non-TLS TACACS+ servers. Operators MUST follow the recommendation of Section 5.1.1 and deploy separate TACACS+ servers for these Non-TLS clients from those used for the TLS clients.

7. IANA Considerations

The authors request that, when this draft is accepted by the working group, the OPSAWG Chairs submit a request to IANA for an early allocation, per [RFC4020] and [RFC6335], of a new well-known system TCP/IP port number for the service name "tacacss" (referenced in this document also as "TACACS+ TLS well-known port ([TBD])"), described as "TACACS+ over TLS". The service name "tacacss" follows the common practice of appending an "s" to the name given to the non-TLS well-

known port name. This allocation is justified in Section 5.3.

This document requests IANA to add a new entry from the "Service Name and Transport Protocol Port Number Registry"
available at <https://www.iana.org/assignments/service-names-port-numbers/>.

Service Name: tacacss
Port Number: tbd
Transport Protocol: TBC
Description: tbc
Assignee: IESG
Contact: IETF Chair
Reference: This-Document

RFC EDITOR: this port number should replace "[TBD]" and the service name should replace "[TBDN]" within this document.

8. Acknowledgments

The author(s) would like to thank Russ Housley, Steven M. Bellovin, Stephen Farrell, Alan DeKok, Warren Kumari, Tom Petch and Mohamed Boucadair for their support, insightful review, and/or comments. [RFC5425] was also used as a basis for the approach to TLS.

9. Normative References

[BCP14] Best Current Practice 14,
<https://www.rfc-editor.org/bcp/bcp14.txt>.
At the time of writing, this BCP comprises the following:

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5425] Miao, F., Ed., Ma, Y., Ed., and J. Salowey, Ed., "Transport Layer Security (TLS) Transport Mapping for Syslog", RFC 5425, DOI 10.17487/RFC5425, March 2009, <https://www.rfc-editor.org/info/rfc5425>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011,

<https://www.rfc-editor.org/info/rfc6151>.

[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, <https://www.rfc-editor.org/info/rfc7301>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8773] Housley, R., "TLS 1.3 Extension for Certificate-Based Authentication with an External Pre-Shared Key", RFC 8773, DOI 10.17487/RFC8773, March 2020, <https://www.rfc-editor.org/info/rfc8773>.

[RFC8907] Dahm, T., Ota, A., Medway Gash, D.C., Carrel, D., and L. Grant, "The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol", RFC 8907, DOI 10.17487/RFC8907, September 2020, <https://www.rfc-editor.org/info/rfc8907>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

10. Informative References

[I-D.ietf-tls-external-psk-guidance] Housley, R., Hoyland, J., Sethi, M., and C. A. Wood, "Guidance for External Pre-Shared Key (PSK) Usage in TLS", Work in Progress, Internet-Draft, draft-ietf-tls-external-psk-guidance-06, 4 February 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-external-psk-guidance-06>.

[RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, DOI 10.17487/RFC3365, August 2002, <https://www.rfc-editor.org/info/rfc3365>.

[RFC4020] Kompella, K. and A. Zinin, "Early IANA Allocation of Standards Track Code Points", RFC 4020, DOI 10.17487/RFC4020, February 2005, <https://www.rfc-editor.org/info/rfc4020>.

[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, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

Commenté [BMI19]: Move to the informative References Section

Commenté [BMI20]: Seems this is not listed in the document

Commenté [BMI21]: Obsoleted by RFC 9325 - Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) (ietf.org)

Commenté [BMI22]: Not cited in the doc

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <https://www.rfc-editor.org/info/rfc7250>.

[RFC7605] Touch, J., "Recommendations on Using Assigned Transport Port Numbers", BCP 165, RFC 7605, DOI 10.17487/RFC7605, August 2015, <https://www.rfc-editor.org/info/rfc7605>.

[TLSCSREC] IANA, "Transport Layer Security (TLS) Parameters", <https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4>.

Commenté [BMI23]: Not cited in the document

Authors' Addresses

Thorsten Dahm
Email: thorsten.dahm@gmail.com

John Heasley
NTT
Email: heas@shrubbery.net

Douglas C. Medway Gash
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
United States of America
Email: dcmgash@cisco.com

Andrej Ota
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America
Email: andrej@ota.si