Operations and Management Area Working Group
Internet-Draft
Updates: 8907 (if approved)
Intended status: Standards Track
Expires: 5 October 2025

T. Dahm

J. Heasley
NTT
D.C. Medway Gash
Cisco Systems, Inc.
A. Ota
Google Inc.

3 April 2025

Terminal Access Controller Access-Control System Plus (TACACS+) over TLS 1.3 (TLS TACACS+)

draft-ietf-opsawg-tacacs-tls13-19

Abstract

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

This document updates RFC 8907~~RFC8907~~.

Commenté [MB1]: For consistency with rfc8907

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

Copyright Notice

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 TACACS+ servers. The protocol provides authentication, authorization and accounting services (AAA) for TACACS+ clients within the device administration use case.

While the content of the protocol is highly sensitive, TACACS+ lacks effective confidentiality, integrity, and authentication of the connection and network traffic between the TACACS+ server and client,

Commenté [MB2]: Use a shorter name to fit

requiring secure transport to safeguard a deployment. The ~~existing~~ TACACS+ mechanisms, ~~are extremely weak~~ as described in Section 10 of [RFC8907]~~.~~, are extremely weak.

Commenté [MB3]: Won't age well.

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

Commenté [MB4]: Avoid confusion with TLS

2. Technical Definitions

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

Obfuscation: TACACS+ was originally intended to incorporate a mechanism for securing the body of its packets. The algorithm is categorized as Obfuscation in Section 10.5.2 of [RFC8907]. The term is used to ensure that the algorithm is not mistaken for encryption. It should not be considered secure.

Non-TLS ~~Connection~~connection: This term refers to the connection defined in [RFC8907]. It is a connection without TLS and, therefore, ~~therefore~~ would be using the unsecure TACACS+ authentication and obfuscation (or the totally unobfuscated option for test). The use of well-known TCP/IP host port number 49 is specified as the default for Non-TLS connections.

TLS ~~Connection~~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.

TLS TACACS+ s~~S~~erver: This document describes a variant of the TACACS+ server, introduced in Section 3.2 of [RFC8907], which ~~utilises~~utilizes TLS for transport, and makes some associated protocol ~~optimisations~~optimizations. Both server variants respond to TACACS+ traffic, but ~~we~~ this document specifically defines a TACACS+ server (whether TLS or Non-TLS) as being bound to specific port number on a particular IP address or hostname. This definition is important in the context of the configuration of TACACS+ clients, to ensure they direct their traffic to the correct TACACS+ servers.

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

3. TACACS+ over TLS

TACACS+ over TLS takes the protocol defined in [RFC8907], removes the option for MD5 obfuscation, and specifies that TLS 1.3 be used for transport (Section 3.1 elaborates TLS version support). A new well-known default host port number is used. The next sections provide further details and guidance.

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

1. Confidentiality and Integrity: The MD5 algorithm underlying the obfuscation mechanism specified in [RFC8907] has been shown to be insecure [RFC6151] when used for encryption. This prevents TACACS+ being used in a [FIPS-140-3~~] ~~]-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 shared secrets of obfuscation for mutual authentication.

The design adheres to the recommendation in [I-D.ietf-uta-require-tls13].

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 (Section 5.3), TLS TACACS+ servers MUST be deployed on a separate TCP/IP port number from Non-TLS TACACS+ servers (preferably on a separate host, as recommended in Section 5.1.1). Because of the widespread use of default port number settings in numerous existing TACACS+ client configurations, a well-known system TCP/IP port number is assigned: the designated port number is [TBD] (Section 7) with the service name ~~[~~(tacacss~~TBDN] ~~) (Section 7). This ensures that the client can separate TLS and Non-TLS ~~non-TLS ~~traffic even where default port numbers are omitted from its TACACS+ server connection configuration.

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 have to be considered by operators.

3.2. TLS Connection

A TACACS+ client initiates a TLS connection by making a TCP connection to a configured TLS TACACS+ server on the TACACS+ TLS port number~~ ([TBD])~~ (Section 7). Once the TCP connection is established, the client MUST immediately begin the TLS negotiation before sending any TACACS+ protocol data.

TLS 1.3 [RFC8446] must be used for transport, though it is expected that TACACS+ as described in this document will work with future

Commenté [MB5]: To cover both default and any other configured number versions of TLS. Earlier versions of TLS MUST NOT be used.

Once the TLS connection is established, the exchange of TACACS+ data proceeds as defined in [RFC8907], except that it is transmitted over TLS as TLS application data and without TACACS+ obfuscation (Section 4). ~~(Section 4)~~

The connection persists until the TLS TACACS+ ~~server or client~~peer closes it, either due to an error, or at the conclusion of the TACACS+ session, or, if Single Connection Mode (Section 4.3 of [RFC8907]) has been negotiated, when an inactivity timeout occurs. Why it closed has no bearing on TLS resumption, unless closed by a TLS error, in which case the ticket might be invalidated.

Commenté [MB6]: Simplify and use the term we have for this.

TACACS+ connections are not long-lived. Non 'single-connect' mode connections are closed as soon as the TACACS+ session completes. Single-connect mode connections are longer lived, but even these are timed out and closed after a short period of inactivity. For this reason, keepalives are not required to be supported.

TACACS+ clients and servers widely support IPv6 configuration in addition to IPv4. This document makes no changes to recommendations in this area.

3.3. TLS Authentication Options

Implementations MUST support ~~certificate~~ certificate-based mutual authentication, to provide a core option for interoperability between deployments. This authentication option is specified in Section 3.4.

In addition to certificate-based TLS authentication, implementations MAY support the following alternative authentication mechanisms:

* Pre-Shared Keys (PSKs) (Section 3.5), also known as external PSKs in TLS 1.3.

* Raw Public Keys (RPKs). The details of RPKs are considered out-of-scope for this document. ~~Please r~~Refer to [RFC7250] and Section 4.4.2 of [RFC8446] for implementation, deployment, and security considerations.

3.4. TLS Certificate-Based Authentication

Commenté [MB7]: Also consistent with rfc5425

TLS certificate authentication is the primary authentication option for TACACS+ over TLS. This section covers certificate-based authentication only.

Deploying TLS ~~C~~certificate-based ~~Authentication~~ authentication correctly will considerably improve the security of TACACS+ deployments. It is essential for implementers and operators to understand the implications of a TLS ~~Certificate~~ certificate-based ~~Authentication~~ authentication solution, including the correct handling of certificates, Certification Authorities (CAs), and all elements of TLS configuration. For

Commenté [MB8]: Use consistent terms, also this is aligned with 8446

Commenté [MB9]: Expand at first use

guidance, please start with [BCP195].

Commenté [MB10]: Cite as a reference

Each peer MUST validate the certificate path of ~~the ~~its remote peer, including revocation checking, as described in Section 3.4.1.

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 NOT permit connection of any peer that presents an invalid TLS ~~Certificate~~certificate.

3.4.1. TLS Certificate Path Verification

The implementation of ~~certificate ~~certificate-based mutual authentication MUST support certificate path verification as described in Section 6 of [RFC5280].

In some deployments, a peer ~~could ~~may be isolated from a remote peer's ~~Certification Authority (~~CA~~)~~. Implementations for these deployments 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.

TLS Cached Information Extension [RFC7924] SHOULD be implemented. This MAY be augmented with ~~Raw Public Keys~~RPKs [RFC7250], though revocation must be handled as it is not part of the standard.

Commenté [MB11]: Already introduced

Other approaches may be used for loading the intermediate certificates onto the client, but MUST include support for revocation checking. For example, [RFC5280] details the ~~AIA (~~Authority Information Access (AIA) extension to provide information about the issuer of the certificate in which the extension appears. It can be used to provide the address of the Online Certificate Status Protocol (OCSP) responder from where revocation status of the certificate (which includes the extension) can be checked.

3.4.2. TLS Certificate Identification

For the client-side validation of presented TLS TACACS+ server identities, implementations MUST follow [RFC9525] validation techniques. Identifier types DNS-ID, IP-ID~~ or~~, or SRV-ID are applicable for use with the TLS TACACS+ protocol, selected by operators depending upon the deployment design. TLS TACACS+ does not use URI-IDs for TLS TACACS+ server identity verification. The wildcard character MUST NOT be included in the presented TLS TACACS+ server identities.

For the TLS TACACS+ server-side validation of client identities, implementations MUST support the ability to configure which fields of a certificate are used for client identification, to verify that the client is a valid source for the received certificate and that it is permitted access to TACACS+. Implementations MUST support either:

Network address based validation methods as described in Section 5.2 of [RFC5425].

or

Client Identity validation of a shared identity in the certificate subjectAltName. This is applicable in deployments where the client securely supports an identity which is shared with the TLS TACACS+ server. Matching of dNSName and iPAddress MUST be supported~~, ~~. ~~other~~Other options defined in Section 4.2.1.6 of [RFC5280] MAY be supported. This approach allows a client's network location to be reconfigured without issuing a new client certificate.

Implementations MUST support the TLS Server Name Indication extension (SNI) (Section 3 of [RFC6066]). TLS TACACS+ clients MUST support the ability to configure the TLS TACACS+ server's domain name, so that it may be included in the SNI "server_name" extension of the client hello. ~~(~~This is distinct from the IP Address or hostname configuration used for the TCP connection~~)~~. ~~See~~ Refer to Section 5.1.5 for security related operator considerations.

Certificate provisioning is out of scope of this document.

3.4.3. Cipher Suites Requirements

Implementations MUST support the TLS 1.3 mandatory cipher suites (Section 9.1 of [RFC8446]). Readers should refer to [BCP195]. The cipher suites offered or accepted SHOULD be configurable so that operators can adapt.

3.5. TLS PSK Authentication

As an alternative to ~~Certificate~~ certificate-based authentication, implementations MAY support ~~Pre-Shared Keys (~~PSKs~~)~~, also known as External PSKs in TLS 1.3 [RFC8446]. These should not be confused with resumption PSKs.

Commenté [MB12]: Already introduced

The use of External PSKs is less well established than certificate-based authentication. It is RECOMMENDED that systems follow the directions of [RFC9257] and Section 4 of [RFC8446]~~, and~~. ~~[RFC9257]~~

Where PSK Authentication is implemented, PSK lengths of at least 16 octets MUST be supported.

PSK Identity MUST follow recommendations of Section 6.1 of [RFC9257]. Implementations MUST support PSK identities of at least 16 octets.

Although this document removes the option of MD5 obfuscation (Section 4), it is still possible that the TLS and ~~non-TLS~~Non-TLS versions of TACACS+ may exist in an ~~organisation~~organization, for example, during migration (~~see~~ Section 6.1). In such cases, the shared secrets configured for TACACS+ obfuscation clients MUST NOT be the same as the PSKs configured for TLS clients.

3.6. TLS Resumption

The TLS Resumption protocol, detailed in [RFC8446], can minimize the number of round trips required during the handshake process. If a TLS client holds a ticket previously extracted from a NewSessionTicket message from the TLS TACACS+ server, it can use the PSK identity tied to that ticket. If the TLS TACACS+ server consents, the resumed session is acknowledged as authenticated and securely linked to the initial session.

The client SHOULD use resumption when it holds a valid unused ticket from the TLS TACACS+ server, as each ticket is intended for a single use only and will be refreshed during resumption. The TLS TACACS+ server can reject a resumption request, but the TLS TACACS+ server SHOULD allow resumption ~~as long as~~if the ticket in question has not expired and has not been used before.

When a TLS TACACS+ server is presented with a resumption request from the TLS client, it MAY still choose to require a full handshake. In this case, the negotiation proceeds as if the session was a new authentication, and the resumption attempt is ignored. As described in Appendix C.4 of [RFC8446], reuse of a ticket allows passive observers to correlate different connections. TLS TACACS+ clients and servers SHOULD follow the client tracking preventions in Appendix C.4 of [RFC8446].

When processing TLS resumption, certificates must be verified to check for revocation during the period since the last NewSessionTicket Message.

The resumption ticket_lifetime SHOULD be configurable, including a zero seconds lifetime. ~~Please r~~Refer to Section 4.6.1 of [RFC8446] for guidance on ticket lifetime.

4. Obsolescence of TACACS+ Obfuscation

[RFC8907] describes the obfuscation mechanism, documented in Section 5.2 of [RFC5425]. Such a 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 TLS TACACS+ server that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG not set (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 Section 4.5 of [RFC8907] Data Obfuscation for TAC_PLUS_UNENCRYPTED_FLAG or key mismatches.

A TACACS+ client that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG not set to 1 ~~(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 TLS TACACS+ 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.

[BCP195] 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 [BCP195] or its subsequent versions.

This document outlines additional restrictions permissible under [BCP195] 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.

This document concerns the use of TLS as transport for TACACS+, and does not make any changes to the core TACACS+ protocol, other than the direct implications of deprecating obfuscation. Operators MUST be cognizant of the security implications of the TACACS+ protocol itself. Further documents are planned, for example, to address the security implications of password based authentication and enhance the protocol to accommodate alternative schemes.

5.1.1. TLS Use

New TACACS+ production deployments SHOULD use TLS authentication and encryption. Also see [RFC3365].

TLS TACACS+ servers (as defined in Section 2) MUST NOT allow Non-TLS connections, because of the threat of downgrade attacks or misconfiguration described in Section 5.3. Instead, separate Non-TLS TACACS+ servers SHOULD be set up to cater for these clients.

It is NOT RECOMMENDED that TLS TACACS+ servers and Non-TLS TACACS+ servers be deployed on the same host, for reasons discussed in Section 5.3. Non-TLS connections would be better served by deploying the required Non-TLS TACACS+ servers on separate hosts.

TACACS+ Clients MUST NOT fail back to a Non-TLS connection if a TLS connection fails. This prohibition includes during the migration of a deployment (~~see~~ Section 6.1).

5.1.2. TLS 0-RTT

TLS 1.3 resumption and PSK techniques make it possible to send Early Data, aka. 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 TLS TACACS+ servers MUST abruptly disconnect clients that do.

5.1.3. TLS Options

Recommendations in [BCP195] MUST be followed~~, in order~~ to determine which TLS versions and algorithms should be supported, deprecated, obsoleted, or abandoned.

Also, Section 9 of [RFC8446] prescribes mandatory supported options.

5.1.4. Unreachable Certification Authority (CA)

Operators should be cognizant of the potential of TLS TACACS+ server and/or 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. The approach mentioned in Section 3.4.1 can help address this~~,~~ and should be considered where implemented.

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 Section 11.1 of [RFC6066].

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-TLS will be used between the TACACS+ client and the TACACS+ server.

This document recommends the use of a separate port number that TLS 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 number 49.

* for TLS connection TACACS+: (TBD).

Implementors may offer a single option for TACACS+ clients and servers to disable all Non-TLS TACACS+ operations. When enabled on a TACACS+ server, it will not respond to any requests from 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 Number

A new port number 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

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

Commenté [MB13]: As the other bullets

However, co-existence of inferior authentication and obfuscated, whether ~~an~~ a 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~~ mitigation to address exposure from Non-TLS connection methods is to refuse Non-TLS connections at the host entirely, perhaps using separate hosts for Non-TLS connections and TLS.

Another approach is mutual configuration that requires TLS. TACACS+ ~~Clients~~ 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. Operational 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 5.1.5. However, it is important that the entire Section 5 is observed.

6.1. Migration

~~In~~ Section 5.2~~, it is mentioned~~ mentions 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:

* T~~t~~he period where any client is configured with both TLS and Non-TLS TACACS+ servers should be minimized.

* ~~T~~The operator must consider the impact of mixed TLS and Non-TLS on security, as mentioned above.

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 Non-TLS TACACS+ servers for these Non-TLS clients from those used for the TLS clients.

6.3. YANG Model for TACACS+ Clients

[I-D.ietf-opsawg-secure-tacacs-yang] specifies a YANG model for managing TACACS+ clients, including TLS support.

7. IANA Considerations

IANA ~~has allocated~~is requested to allocate a new well-known system TCP/IP port number ([TBD]) for the service name "tacacss"~~, per [RFC4020] and [RFC6335]~~, 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.

IANA ~~has~~ is requested to added~~ed~~ (TBD) a new ~~entity~~ entry to the "Service name and Transport Protocol Port Number Registry" available at https://www.iana.org/assignments/service-names-port-numbers/

Commenté [MB14]: Transform to a reference entry

Service Name: tacacss

Port Number: [TBD]

Transport Protocol: TCP

Description: TLS Secure Login Host Protocol (TACACSS)

Assignee: IESG

Contact: IETF Chair

Reference: [TBDN] (This Document)

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

Considerations about service discovery are out of scope of this document.

8. Acknowledgments

The author(s) would like to thank Russ Housley, Steven M. Bellovin, Stephen Farrell, Alan DeKok, Warren Kumari, Tom Petch, Tirumal Reddy, Valery Smyslov, and Mohamed Boucadair for their support, insightful review, and/or comments. [RFC5425] was also used as a basis for the general approach to TLS. [RFC9190] was used as a basis for TLS Resumption Recommendations. draft-ietf-radext-tls-psk, although still in draft form at the time of writing, was used as a model for PSK Recommendations.

9. Normative References

[BCP195] Best Current Practice 195, <https://www.rfc-editor.org/info/bcp195>. At the time of writing, this BCP comprises the following:

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>.

[RFC2119] 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>.

[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>.

[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>.

[RFC7924] Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", RFC 7924, DOI 10.17487/RFC7924, July 2016, <https://www.rfc-editor.org/info/rfc7924>.

[RFC8174] 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>.

[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>.

[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>.

[RFC9525] Saint-Andre, P. and R. Salz, "Service Identity in TLS", RFC 9525, DOI 10.17487/RFC9525, November 2023, <https://www.rfc-editor.org/info/rfc9525>.

10. Informative References

Commenté [MB15]: Clean unused references

[FIPS-140-3] National Institute of Standards and Technology, U.S. Department of Commerce, "NIST Federal Information Processing Standards (FIPS) Publication 140-3", <https://csrc.nist.gov/pubs/fips/140-3/final>.

[ietf-opsawg-secure-tacacs-yang] Boucadair, M., Ed., Wu, B., Zheng, G., and M. Wang, "A YANG Data Model for Terminal Access Controller Access-Control System Plus (TACACS+)", <https://datatracker.ietf.org/doc/draft-ietf-opsawg-secure-tacacs-yang/>.

[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>.~~

[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>.

~~[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>.~~

[RFC9190] Preuß Mattsson, J. and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3", RFC 9190, DOI 10.17487/RFC9190, February 2022, <https://www.rfc-editor.org/info/rfc9190>.

[RFC9257] Housley, R., Hoyland, J., Sethi, M., and C. A. Wood, "Guidance for External Pre-Shared Key (PSK) Usage in TLS", RFC 9257, DOI 10.17487/RFC9257, July 2022,

<https://www.rfc-editor.org/info/rfc9257>.

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

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