Stream: Internet Engineering Task Force (IETF)
RFC: 9887
Updates: 8907
Category: Standards Track
Published: December 2025
ISSN: 2070-1721
Authors: T. Dahm, J. Heasley (*NTT*), D.C. Medway Gash (*Cisco Systems, Inc.*), A. Ota (*Google Inc.*)

RFC 9887
Terminal Access Controller Access-Control System Plus (TACACS+) over TLS 1.3

Abstract

This document specifies the use of Transport Layer Security (TLS) version 1.3 to secure the communication channel between a Terminal Access Controller Access-Control System Plus (TACACS+) client and server. TACACS+ is a protocol used for Authentication, Authorization, and Accounting (AAA) in networked environments. The original TACACS+ protocol does not mandate the use of encryption or secure transport. This specification defines a profile for using TLS 1.3 with TACACS+, including guidance on authentication, connection establishment, and operational considerations. The goal is to enhance the confidentiality, integrity, and authenticity of TACACS+ traffic, aligning the protocol with modern security best practices.

This document updates RFC 8907.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9887.

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.

The content of the protocol is highly sensitive and requires secure transport to safeguard a deployment. However, TACACS+ lacks effective confidentiality, integrity, and authentication of the connection and network traffic between the TACACS+ server and client. The security mechanisms as described in Section 4.5 of [RFC8907] are extremely weak.

To address these deficiencies, this document updates the TACACS+ protocol to use TLS 1.3 authentication and encryption [RFC8446], and obsoletes the use of TACACS+ obfuscation mechanisms. The maturity of TLS in version 1.3 and above makes it a suitable choice for the TACACS+ protocol.

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

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.

TLS TACACS+ server: This document describes a variant of the TACACS+ server, introduced in Section 3.2 of [RFC8907], which utilizes TLS for transport, and makes some associated protocol optimizations. Both server variants respond to TACACS+ traffic, but this document specifically defines a TACACS+ server (whether TLS or non-TLS) as being bound to a 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.

2.1. 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.

3. TACACS+ over TLS

TACACS+ over TLS takes the protocol defined in [RFC8907], removes the option for obfuscation, and specifies that TLS 1.3 be used for transport (Section 3.1 elaborates on 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+ from being used in a deployment compliant with [FIPS-140-3]. Securing the 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.

This document adheres to the recommendations in [REQ-TLS13].

3.1. Separating TLS Connections

Peers implementing the TACACS+ protocol variant defined in this document **MUST** apply mutual authentication and encrypt all data exchanged between them. Therefore, when a TCP connection is established for the service, a TLS handshake begins immediately. Options that upgrade an initial non-TLS connection **MUST NOT** be used; see Section 5.3.

To ensure clear separation between TACACS+ traffic using TLS and that which does not (see Section 5.3), servers supporting TACACS+ over TLS **MUST** listen on a TCP/IP port distinct from that used by non-TLS TACACS+ servers. It is further **RECOMMENDED** to deploy the TLS and non-TLS services on separate hosts, as discussed in Section 5.1.1.

Given the prevalence of default port usage in existing TACACS+ client implementations, this specification assigns the well-known TCP port number 300 for TACACS+ over TLS (see Section 7).

While the use of the designated port number is strongly encouraged, deployments with specific requirements **MAY** use alternative TCP port numbers. In such cases, operators must carefully consider the operational implications described in Section 5.3.

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. Once the TCP connection is established, the client **MUST** immediately begin the TLS negotiation before sending any TACACS+ protocol data.

A minimum of TLS 1.3 [RFC8446] **MUST** be used for transport. It is expected that TACACS+, as described in this document, will work with future versions of TLS. Earlier versions of TLS **MUST NOT** be used.

Once the TLS connection has been successfully established, the exchange of TACACS+ data **MUST** proceed in accordance with the procedures defined in [RFC8907]. However, all TACACS+ messages **SHALL** be transmitted as TLS application data. The TACACS+ obfuscation mechanism defined in [RFC8907] **MUST NOT** be applied when operating over TLS (Section 4).

TLS TACACS+ connections are generally not long-lived. The connection will be closed by either a peer if it encounters an error or an inactivity timeout.

For connections not operating in Single Connection Mode (as defined in Section 4.3 of [RFC8907]), the TCP session **SHALL** be closed upon completion of the associated TACACS+ session. Connections operating in Single Connection Mode **MAY** persist for a longer duration but are typically subject to timeout and closure after a brief period of inactivity. Consequently, support for transport-layer keepalive mechanisms is not required.

Why a connection is closed has no bearing on TLS resumption, unless closed by a TLS error, in which case it is possible that the ticket has been invalidated.

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-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. Refer to [RFC7250] and Section 4.4.2 of [RFC8446] for implementation, deployment, and security considerations.

3.4. TLS Certificate-Based Authentication

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

Deploying TLS certificate-based 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-based authentication solution, including the correct handling of certificates, Certificate Authorities (CAs), and all elements of TLS configuration. For guidance, start with [BCP195].

Each peer **MUST** validate the certificate path of 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.

3.4.1. TLS Certificate Path Verification

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

In some deployments, a peer may be isolated from a remote peer's CA. Implementations for these deployments **MUST** support certificate chains (aka bundles or chains of trust), where the entire chain of the remote peer's certificate is stored on the local peer.

TLS Cached Information Extension [RFC7924] **SHOULD** be implemented. This **MAY** be augmented with RPKs [RFC7250], though revocation must be handled as it is not part of that specification.

Other approaches may be used for loading the intermediate certificates onto the client, but they **MUST** include support for revocation checking. For example, [RFC5280] details the 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 the 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 the validation techniques defined in [RFC9525]. Identifier types DNS-ID, IP-ID, or SRV-ID are applicable for use with the TLS TACACS+ protocol; they are selected by operators depending upon the deployment design. TLS TACACS+ does not use URI-IDs for TLS TACACS+ server identity verification.

Wildcards in TLS TACACS+ server identities simplify certificate management by allowing a single certificate to secure multiple servers in a deployment. However, this introduces security risks, as compromising the private key of a wildcard certificate impacts all servers using it. To address these risks, the guidelines in Section 6.3 of [RFC9525] **MUST** be followed, and the wildcard **SHOULD** be confined to a subdomain dedicated solely to TACACS+ servers.

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 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 (SNI) extension (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). 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-based authentication, implementations **MAY** support PSKs, also known as external PSKs in TLS 1.3 [RFC8446]. These should not be confused with resumption PSKs.

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].

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 obfuscation (Section 4), it is still possible that the TLS and non-TLS versions of TACACS+ exist in an organization, for example, during migration (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

TLS Resumption [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 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. Refer to Section 4.6.1 of [RFC8446] for guidance on ticket lifetime.

4. Obsolescence of TACACS+ Obfuscation

The obfuscation mechanism documented in Section 4.5 of [RFC8907] is weak.

The introduction of TLS authentication and encryption to TACACS+ replaces this former mechanism, so obfuscation is hereby obsoleted. This section describes how the TACACS+ client and servers **MUST** operate regarding 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 bit set to 1.

A TLS TACACS+ server that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG bit not set to 1 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 bit set to 1, and terminate the session. This behavior corresponds to that defined in Section 4.5 of [RFC8907] regarding data obfuscation for TAC_PLUS_UNENCRYPTED_FLAG or key mismatches.

A TACACS+ client that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG bit not set to 1 **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 and deploying protocols that use TLS. 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 (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.

TLS TACACS+ clients and servers **MUST NOT** include the "early_data" extension. See Sections 2.3 and 4.2.10 of [RFC8446] for security concerns.

5.1.3. TLS Options

Recommendations in [BCP195] **MUST** be followed 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 Certificate 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.

5.1.5. TLS Server Name Indicator (SNI)

Operators should be aware that the TLS SNI extension is part of the TLS client hello, which is sent in cleartext. It is, therefore, subject to eavesdropping. Also see Section 11.1 of [RFC6066].

5.1.6. Server Identity Wildcards

The use of wildcards in TLS server identities creates a single point of failure: a compromised private key of a wildcard certificate impacts all servers using it. Their use **MUST** follow the recommendations of Section 7.1 of [RFC9525]. Operators **MUST** ensure that the wildcard is limited to a subdomain dedicated solely to TLS TACACS+ servers. Further, operators **MUST** ensure that the TLS TACACS+ servers covered by a wildcard certificate are impervious to redirection of traffic to a different server (for example, due to on-path attacks or DNS cache poisoning).

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 port values:

- 49: for non-TLS connection TACACS+
- 300: for TLS connection TACACS+

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 (rather than a mechanism that negotiates an upgrade from an initial non-TLS TACACS+ connection) 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 on-path attacks that can interfere with upgrade, and
- prevention of the accidental exposure of sensitive information due to misconfiguration.

However, the coexistence of inferior authentication and obfuscation, whether a non-TLS connection or deprecated parts that compose TLS, also presents an opportunity for downgrade attacks. Causing failure of connections to the TLS-enabled service or the negotiation of shared algorithm support are two such downgrade attacks.

The simplest mitigation 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 and servers **SHOULD** support configuration that requires peers, globally and individually, to 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.

It is essential for operators to understand the implications of a TLS certificate-based authentication solution, including the correct handling of certificates, CAs, and all elements of TLS configuration. Refer to [BCP195] for guidance. Attention is drawn to the provisioning of certificates to all peers, including TACACS+ TLS clients, to permit the mandatory mutual authentication.

6.1. Migration

Section 5.2 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:

- The period where any client is configured with both TLS and non-TLS TACACS+ servers should be minimized.
- The operator must consider the security impact of supporting both TLS and non-TLS connections, 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

[TACACS-YANG] specifies a YANG model for managing TACACS+ clients, including TLS support.

7. IANA Considerations

IANA has allocated the following new well-known system in the "Service Name and Transport Protocol Port Number Registry" (see <https://www.iana.org/assignments/service-names-port-numbers>). The service name "tacacss" follows the common practice of appending an "s" to the name given to the non-TLS well-known port name. See the justification for the allocation in Section 5.3.

Service Name: tacacss
Port Number: 300
Transport Protocol: TCP
Description: TLS Secure Login Host Protocol (TACACSS)
Assignee: IESG
Contact: IETF Chair
Reference: RFC 9887

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. Although still in draft form at the time of writing, [RFC9813] was used as a model for PSK recommendations.

9. References

9.1. Normative References

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

Moriarty, K. and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1", BCP 195, RFC 8996, DOI 10.17487/RFC8996, March 2021, <https://www.rfc-editor.org/info/rfc8996>.

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>.

9.2. Informative References

[FIPS-140-3] NIST, "Security Requirements for Cryptographic Modules", NIST FIPS 140-3, DOI 10.6028/NIST.FIPS.140-3, March 2019, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf>.

[REQ-TLS13] Salz, R. and N. Aviram, "New Protocols Using TLS Must Require TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-uta-require-tls13-12, 14 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-uta-require-tls13-12>.

[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>.

[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>.

[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>.

[RFC9813] DeKok, A., "Operational Considerations for Using TLS Pre-Shared Keys (TLS-PSKs) with RADIUS", BCP 243, RFC 9813, DOI 10.17487/RFC9813, July 2025, <https://www.rfc-editor.org/info/rfc9813>.

[TACACS-YANG] Boucadair, M. and B. Wu, "A YANG Data Model for Terminal Access Controller Access-Control System Plus (TACACS+)", Work in Progress, Internet-Draft, draft-ietf-opsawg-secure-tacacs-yang-13, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-secure-tacacs-yang-13>.

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