

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Technical Definitions](#)
 - [2.1. Unsecure Connection](#)
 - [2.2. Peer](#)
 - [2.3. TLS Connection](#)
 - [2.4. Obfuscation](#)
- [3. TLS for TACACS+](#)
 - [3.1. Well-Known TCP/IP Port](#)
 - [3.2. TLS Connection](#)
 - [3.2.1. Cipher Requirements](#)
 - [3.2.2. TLS Authentication](#)
 - [3.3. TLS Identification](#)
- [4. Obsolescence of TACACS+ Obfuscation](#)
- [5. Security Considerations](#)
 - [5.1. TLS](#)
 - [5.1.1. TLS Use](#)
 - [5.1.2. TLS 0-RTT](#)
 - [5.1.3. TLS PSK](#)
 - [5.1.4. TLS Options](#)
 - [5.1.5. Unreachable TLS CA](#)
 - [5.2. Well-Known TCP/IP Port](#)
- [6. IANA Considerations](#)
- [7. Acknowledgments](#)
- [8. Normative References](#)
- [9. Informative References](#)
- [Authors' Addresses](#)

1. Introduction

The [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 for TACACS+ clients.

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. The existing mechanisms of TACACS+ are extremely weak and the Security Considerations section of the [TACACS+ Protocol \[RFC8907\]](#) adequately describes this.

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

2. Technical Definitions

The Technical Definitions section of the [TACACS+ Protocol \[RFC8907\]](#) is fully applicable here and will not be repeated, though might be augmented. The following terms are also used in this document.

2.1. Unsecure Connection

This is another term for a Connection as defined in [TACACS+ Protocol \[RFC8907\]](#). It is a Connection without TLS and therefore being plaintext or possibly using unsecure TACACS+ authentication and obfuscation.

2.2. Peer

This refers to a TACACS+ Server or Client.

2.3. TLS Connection

A TLS Connection is a TCP/IP connection with TLS authentication and encryption used by TACACS+ for transport, similar to a Connection as defined in [TACACS+ Protocol \[RFC8907\]](#).

2.4. Obfuscation

Inferior form of encryption used in TACACS+, referred to as obfuscation in [\[RFC5425\]](#), [Section 5.2](#) to indicate that it is not encryption and is utterly insufficient.

3. TLS for TACACS+

TACACS+ connections are TCP/IP connections initiated by the Client to the Server. The well-known TCP/IP port 49 on the Server is used for unobfuscated and obfuscated connections as defined in the [TACACS+ Protocol \[RFC8907\]](#). A connection might be used for only a single Session or the multiplexing of multiple Sessions in TACACS+ Single Connection Mode.

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

1. Confidentiality and Integrity: The MD5 obfuscation specified in the original protocol definition is not fit for purpose, requiring that TACACS+ be deployed over a secured network. Securing TACACS+ protocol with TLS is intended to provide confidentiality and integrity without requiring the provision of a secured network.
2. Peer authentication: The use of shared keys to add and remove the MD5 obfuscation was intended to provide a form of Peer authentication for the TACACS+ protocol. This document obsoletes the MD5 obfuscation, and specifies that the authentication capabilities of TLS are used to allow the Peers to authenticate each other.

3.1. Well-Known TCP/IP Port

All data exchanged by TACACS+ Peers MUST be encrypted, including the authentication of the Peers. Therefore, TLS Hello MUST be initiated by the client immediately upon the establishment of the TCP/IP connection.

This document favors the predictable use of TLS security for a deployment, see ([Section 5.2](#)). TACACS+ TLS will therefore follow [[RFC7605](#)], where a different well-known system TCP/IP port is assigned by IANA, port [[TBD](#)] ([Section 6](#)) with the service name [[TBDN](#)] ([Section 6](#)), for TLS connections.

TACACS+ TLS could use any other TCP port by operator configuration, though [Section 5.2](#) should still be considered.

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 ([[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 a 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.

3.2.1. Cipher Requirements

Implementations MUST support the TLS 1.3 mandatory cipher suites (See 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 [[TLSCSREC](#)].

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 this 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 a Peer that offers an invalid TLS Certificate.

3.2.2.1. TLS Certificate Path Verification

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

3.3. TLS Identification

In addition to authentication of TLS certificates, implementations MUST support policy consideration of Peer-identifying certificate fields and policy used 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:

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.

4. Obsolescence of TACACS+ Obfuscation

The original draft of TACACS+ described the Obfuscation mechanism, documented in [[RFC5425](#)], [Section 5.2](#). It is insufficient for modern purposes.

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 (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.

A TACACS+ client that receives a packet with the TAC_PLUS_UNENCRYPTED_FLAG not set (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. This does not in itself protect the server nor clients; the operator and equipment vendors have a role. That role is to diligently follow current best practices for maintaining the integrity of network devices and selection of TLS key and encryption algorithms.

5.1.1. TLS Use

TLS encryption SHOULD be used in deployments when both the Clients and Servers support it. Servers that support TLS encryption MAY be configured to allow Unsecure Connections when TLS encryption is not supported by the Client, but this is NOT RECOMMENDED because of the threat of downgrade attacks, as described in [Section 5.2](#). Unsecure Connections would be better served by separate Servers from the TLS Servers.

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, aka. 0-RTT data, data that is sent before the TLS handshake completes. Replay of this data is possible. Given the sensitivity of TACACS+ data, a Client 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 PSK

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 SHOULD NOT be shared among Clients or Servers to limit exposure of a compromised key and to ease key rotation. Also see [[RFC8773](#)] and [[I-D.ietf-tls-external-psk-guidance](#)].

PSKs are otherwise considered out-of-scope for this document.

5.1.4. TLS Options

Unfortunately, no single and timely TLS recommendations document exists. Therefore, implementers 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. Useful examples are the TLS specifications themselves ([TLS 1.3](#) [[RFC8446](#)]), which prescribes mandatory support in Section 9, and TLS Recommendations [[RFC7525](#)].

5.1.5. Unreachable TLS CA

Operators SHOULD be cognizant of the potential of Server and/or Client isolation from their Peer's Certificate Authority (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. Certificate caching and [Raw](#)

[Public Keys \[RFC7250\]](#) are methods to address this, but both are out of scope for this document. Certificate fingerprints are another option.

5.2. 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 Unsecure 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 the exposure from Unsecure Connection methods is to refuse Unsecure Connections at the server entirely, perhaps using separate servers for Unsecure 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.

Servers and Clients could also maintain a cache of Peers that have engaged in TACACS+ TLS connections and demand TLS from that point forward. However, this has potential to be a Denial of Service (DoS) vector, whereby an attacker causes a server to believe that a client that does not support TLS has successfully connected with TLS.

6. 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 5.2](#).

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

7. Acknowledgments

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

8. Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.
<<https://www.rfc-editor.org/bcp/bcp14.txt>>
- [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>>.
- [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>>.

9. 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://www.ietf.org/archive/id/draft-ietf-tls-external-psk-guidance-06.txt>>.
- [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>>.
- [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>>.

Authors' Addresses

Thorsten Dahm

Email: thorsten.dahm@gmail.com

Douglas Gash

Cisco Systems, Inc.

Email: dcmgash@cisco.com

Andrej Ota

Email: andrej@ota.si

John Heasley
NTT

Email: heas@shrubbery.net