Operations and Management Area Working Group — T. Dahm
Internet-Draft
Updates: RFC8907 (if approved) — D. Gash
Intended status: Standards Track — Cisco Systems, Inc.
Expires: 31 December 2023 — A. Ota

J. Heasley
NTT
29 June 2023

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

draft-ietf-opsawg-tacacs-tls13-03

Abstract

The Terminal Access Controller Access-Control System Plus (TACACS+) p~~TACACS+ P~~rotocol ~~[~~ (RFC 8907~~]~~) provides device administration for routers, network access servers, and other networked computing devices via one or more centralized servers. This document~~, a companion to the TACACS+ protocol [RFC8907],~~ adds Transport Layer Security ~~(currently defined by TLS 1.3 [RFC8446])~~ support to TACACS+ and obsoletes former inferior security mechanisms.

Commenté [BMI1]: https://authors.ietf.org/required-content:

« An abstract should be complete in itself, so it should not contain citations unless they are completely defined within the abstract. Abbreviations appearing in the abstract should follow the Abbreviations guidelines. »

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

Copyright Notice

Table of Contents

1. Introduction

The Terminal Access Controller Access-Control System Plus (TACACS+) p~~TACACS+ P~~rotocol [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 (AAA) 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~~ servers and clients. The existing TACACS+ mechanisms ~~of TACACS+~~ are extremely weak ~~and the~~ ~~Security Considerations section of the TACACS+ Protocol~~as adequately described in Section 10 of [RFC8907] ~~adequately describes this~~.

Commenté [BMI2]: To be consistent with 8907

Commenté [BMI3]: Better to provide the explicit section so for the reader's convenience

To address these deficiencies, this document updates the TACACS+ ~~Protocol~~ 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~~Terms defined in Section 3 of [RFC8907] ~~is~~ are fully applicable here and will not be repeated. The following terms are also used in this document.

2.1. Unsecure Connection

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

2.2. TLS Connection

A TLS ~~Connection~~ connection is a TCP/IP connection with TLS authentication and encryption used by TACACS+ for transport. ~~The~~ A TLS ~~Connection~~ connection for TACACS+ is always between one ~~Client~~ TACACS+ client and one ~~Server~~ TACACS+ server as ~~defined in TACACS+ Protocol [RFC8907]~~.

2.3. Peer

~~In the context of a TLS Connection, the~~ The peer of a TACACS+ ~~Client~~ client (or server) in the context of a connection is the ~~Server~~server (or client).~~, and the peer is the TACACS+ Server is the Client.~~ Together, the ends of a TLS Connection are referred as the peers.

2.4. Obfuscation

Obfuscation is ~~Inferior~~ inferior form of encryption used in TACACS+, referred to as obfuscation in Section 5.2 of [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~~ a ~~Client~~ client to ~~the~~ a ~~Server~~server. By default, the server listens on ~~The~~ 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~~ session or ~~the~~ multiplexing ~~of~~ multiple ~~Sessions~~ sessions ~~in TACACS+ Single Connection Mode~~(a.k.a. TACACS+ Single

Commenté [BMI4]: Obfuscation is not discussed in this spec.

I think you simply need to point to obfuscation form in the base TACACS+ spec.

Commenté [BMI5]: Is it normal that RFC9325 is not mentioned in the doc?

Connection Mode, Section 4.3 of [RFC8907]).

TLS support is ~~introduced~~ added ~~in~~to TACACS+ to fulfill the following requirements:

1. Confidentiality and Integrity: The MD5 obfuscation specified in [RFC8907]~~the original protocol definition~~ is not fit for purpose, requiring that TACACS+ be deployed ~~over a secured network~~in a secured environment. Securing the TACACS+ protocol with TLS is intended to provide confidentiality and integrity ~~without requiring the provision of a secured network~~and relax the deployment constraints imposed in Section 10.2 of [RFC8907].

2. Peer authentication: The use of shared keys to add and remove the MD5 obfuscation was intended to provide a form of ~~Peer~~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~~mutual authentication of peers.

3.1. Well-Known TCP/IP Port Number

All data exchanged by TACACS+ p~~P~~eers 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 8~~)~~. TACACS+ TLS will therefore follow [RFC7605], where a different well-known system TCP/IP port is assigned by IANA, port [TBD] (Section 7) with the service name [TBDN] (Section 7), for TLS connections.

TACACS+ TLS could use any other TCP port by operator configuration, though Section 8 should still be considered.

3.2. TLS Connection

A TACACS+ ~~Client~~ client initiates a TLS connection by making a TCP connection to a configured ~~Server~~ 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~~ server or ~~Client~~ client closes it. It might be closed due to an error or at the conclusion of the TACACS+

Commenté [BMI6]: Simply pointing to this part from the base spec would be sufficient:

« This mechanism MUST NOT be used in modern deployments. It MUST NOT be used outside a secured deployment. »

Commenté [BMI7]: I'm afraid that more justification is needed to ask for well-know port.

For example, why wouldn't a service name be sufficient to discover the port number used in a deployment? Etc.

Commenté [BMI8]: I suspect that the designated expert will ask: If it can be configured, why then we need to burn a new well-known port number.

Commenté [BMI9]: How the server is configured to the client?

Do we allow to configure names? If domain name are allowed, I guess we need to discuss that servers presents the name together with the certificate.

Commenté [BMI10]: No, other ports can be used as per the previous sentence.

Session. If Single Connection Mode has been negotiated, it might remain open after a successful ~~Session~~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 might be invalidated.

3.2.1. Cipher Requirements

Implementations MUST support the TLS 1.3 mandatory cipher suites (See Section 9.1 of ~~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 [TLSCSREC].

Commenté [BMI11]: Instead, https://www.rfc-editor.org/rfc/rfc9325.html#name-recommendations-cipher-suit

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

Commenté [BMI12]: As this is local to an implementation

Unless disabled by configuration, a ~~Peer~~ peer MUST disconnect ~~a~~ the remote ~~Peer~~ peer that ~~offers~~ presents an invalid TLS Certificate.

3.2.2.1. TLS Certificate Path Verification

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

Because a ~~Peer~~ peer could be isolated from a remote ~~Peer's~~ 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.

3.3. TLS Identification

In addition to authentication of TLS certificates, implementations MUST support policy consideration of ~~Peer~~peer-identifying certificate fields and policy used to verify that the p~~P~~eer is a valid source for the received certificate and that it is permitted access to TACACS+. Implementations MUST support either:

Commenté [BMI13]: What does that mean?

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

Commenté [BMI14]: Which part of that section are you referring to?

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.

Commenté [BMI15]: Shouldn’t draft-ietf-uta-rfc6125bis be mentioned? Some discussion about which IDs (DNS-ID, SRV-ID, CN-ID, URI-ID) are recommended for the server identity would be useful.

Commenté [BMI16]: BTW, you may indicate that certificate provisioning is out of scope.

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

Commenté [BMI17]: I would an explicit mention that clients SHOULD include the server domain name in the SNI extension (assuming that domain name is provisioned)?

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.

Commenté [BMI18]: ?

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. This behavior corresponds to that defined in RFC8907 Section 4.5. 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 (cleared), MUST terminate the Session, and SHOULD log this error.

5. Security Considerations

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 follow current best practices for maintaining the integrity of network devices and selection of TLS key and encryption algorithms.

5.1. TLS Options

No single and timely TLS recommendations document exists. Therefore, implementers and operators SHOULD refer to TLS RFCs to ensure the versions are current and which 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].

5.2. 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~~ clients MUST NOT send data until the full TLS handshake completes; that is, ~~Clients~~ clients MUST NOT send 0-RTT data and ~~Servers~~ servers MAY abruptly disconnect c~~C~~lients that do.

5.3. 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~~ clients or s~~S~~ervers 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.

6. Operator Considerations

This section outlines considerations which are specific to operators. It is important that operators ensure their deployments address the considerations in Section 5.

6.1. TLS Use

TLS encryption ~~SHOULD~~ is expected to be used in deployments when both the ~~Clients~~clients and ~~Servers~~ servers support it. In order to prevent downgrade attacks, Servers SHOULD keep separate and disjoint lists of clients supporting TLS and Unsecure Connections. 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].

6.2. Migration to TLS

When ~~Migrating~~ migrating from legacy service to TLS, any mixture of Unsecure Connected Servers and TLS-Protected Servers in the same redundant lists on clients SHOULD be minimised.

Commenté [BMI19]: Actually, that's more subtle :

Section E.5 of [RFC8446] states the following:

Replayable 0-RTT data presents a number of security threats to TLS- using applications, unless those applications are specifically engineered to be safe under replay (minimally, this means idempotent, but in many cases may also require other stronger conditions, such as constant-time response).

...

Application protocols MUST NOT use 0-RTT data without a profile that defines its use. That profile needs to identify which messages or interactions are safe to use with 0-RTT and how to handle the situation when the server rejects 0-RTT and falls back to 1-RTT.

Commenté [BMI20]: In the context of TACACS+?

Commenté [BMI21]: I would sue a strong language here.

Commenté [BMI22]: This is not a reco. TLS can be used only if the peers support it.

Commenté [BMI23]: This is another argument that separate port number is not used.

Commenté [BMI24]: This may conflict with the MUST NOT in RFC8907:

This mechanism MUST NOT be used in modern deployments.

Commenté [BMI25]: I think this is covered if you have a generic statement about whether you adhere to rfc9325

Commenté [BMI26]: Not sure this is concrete enough.

After migration, the production deployment SHOULD NOT mix Legacy and TLS-Protected Servers within Server lists configured on clients.

6.3. Downgrade ~~attacks~~ Attacks in TLS

All clients and servers in a deployment should be configured with consistent algorithm and cypher options (Section 5.1) to prevent harm from downgrade attacks.

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

Commenté [BMI27]: « peer » assumes that a connection is in place, while the configuration happens before. I would simply use clients/servers.

6.4. Unreachable Certificate Authority (CA)

Operators SHOULD be cognizant of the potential of 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. 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.

6.5. 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.

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 8.

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

8. Discussion on Separate port vs Negotiated TLS

Commenté [BMI28]: These are arguments for why a separate port number may be needed, not why a new well-known port is justified.

The authors concluded that a new port is considered superior to negotiation of TLS using "STARTTLS" command because:

* it allows easy blocking the unobfuscated or obfuscated connections by the TCP/IP port number,

- passive Intrusion Detection Systems (IDSs) monitoring the unobfuscated deployments will be unaffected by the introduction of TLS,

- Man in the Middle (MitM) attacks that can interfere with STARTTLS will be avoided

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

9. 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.

10. 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>.

[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>.

[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>.

11. 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>.

[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