Workgroup: RADEXT Working Group
Internet-Draft: draft-dekok-radext-tls-psk-01
Published: 6 July 2023
Intended Status: Informational
Expires: 7 January 2024
Authors: A. DeKok
FreeRADIUS

RADIUS and TLS-PSK

Abstract

This document gives implementation and operational considerations for using TLS-PSK with RADIUS/TLS (RFC6614) and RADIUS/DTLS (RFC7360).

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dekok-radext-tls-psk/.

Discussion of this document takes place on the RADEXT Working Group mailing list (mailto:radext@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/radext/.

Source for this draft and an issue tracker can be found at https://github.com/freeradius/radext-tls-psk.git.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 7 January 2024.

Copyright Notice

Table of Contents

1. Introduction

The previous specifications "Transport Layer Security (TLS) Encryption for RADIUS" [RFC6614] and " Datagram Transport Layer Security (DTLS) as a Transport Layer for RADIUS" [RFC7360] defined how (D)TLS can be used as a transport protocol for RADIUS. However, those documents do not provide guidance for using TLS-PSK with RADIUS. This docoument provides that missing guidance, and gives implementation and operational considerations.

2. Terminology

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.

TBD

3. History

TLS deployments usually rely on certificates in most common uses. However, we recognize that it may be difficult to fully upgrade client implementations to allow for certificates to be used with RADIUS/TLS and RADIUS/DTLS. These upgrades involve not only implementing TLS, but can also require significant changes to administration interfaces and application programming interfaces (APIs) in order to fully support certificates.

For example, unlike shared secrets, certificates expire. This expiration means that a working system using TLS can suddenly stop working. Managing this expiration can require additional notification APIs on RADIUS clients and servers which were previously not required when shared secrets were used.

Certificates also require the use of certification authorities (CAs), and chains of certificates. RADIUS implementations using TLS therefore have to track not just a small shared secret, but also potentially many large certificates. The use of TLS-PSK can therefore provide a simpler upgrade path for implementations to transition from RADIUS shared secrets to TLS.

4. General Discussion of PSKs and PSK Identies.

Before we define any RADIUS-specific use of PSKs, we must first review the current standards for PSKs, and give general advice on PSKs and PSK identies.

The requirements in this section apply to both client and server implementations which use TLS-PSK. Client-specific and server-specific issues are discussed in more detail later in this document.

4.1. Requirements on PSKs

Reuse of a PSK in multiple versions of TLS (e.g. TLS 1.2 and TLS 1.3) is considered unsafe ([RFC8446] Section E.7). Where TLS 1.3 binds the PSK to a particular key deriviation function, TLS 1.2 does not. This binding means that it is possible to use the same PSK in different hashes, leading to the potential for attacking the PSK by comparing the hash outputs. While there are no known insecurities, these uses are not known to be secure, and should therefore be avoided.

[RFC9258] adds a key derivation function to the import interface of (D)TLS 1.3, which binds the externally provided PSK to the protocol version. In particular, that document:

> ... describes a mechanism for importing PSKs derived from external PSKs by including the target KDF, (D)TLS protocol version, and an optional context string to ensure uniqueness. This process yields a set of candidate PSKs, each of which are bound to a target KDF and protocol, that are separate from those used in (D)TLS 1.2 and prior versions. This expands what would normally have been a single PSK and identity into a set of PSKs and identities.

If an implementation supports both TLS 1.2 and TLS 1.3, it MUST require that TLS 1.3 be negotiated in RADIUS/TLS and RADIUS/DTLS. This requirement prevents reuse of a PSK with multiple TLS versions, which prevents the attacks discussed in [RFC8446] Section E.7.

It is RECOMMENDED that systems follow the directions of [RFC9257] Section 4 for the use of external PSKs in TLS. That document provides extremely useful guidance on generating and using PSKs.

Implementations MUST support PSKs of at least 32 octets in length, and SHOULD support PSKs of 64 octets. Implementations MUST require that PSKs be at least 16 octets in length. That is, short PSKs MUST NOT be permitted to be used.

Administrators SHOULD use PSKs of at least 24 octets, generated using a source of cryptographically secure random numbers. Implementors needing a secure random number generator should see [RFC8937] for for further guidance. PSKs are not passwords, and administrators should not try to manually create PSKs.

Passwords are generally intended to be remembered and entered by people on a regular basis. In contrast, PSKs are intended to be entered once, and then automatically saved in a system configuration. As such, due to the limited entropy of passwords, they are not acceptable for use with TLS-PSK, and would only be acceptable for use with a password-authenticated key exchange (PAKE) TLS method.

We also incorporate by reference the requirements of Section 10.2 of [RFC7360] when using PSKs.

4.1.1. Interaction between PSKs and Shared Secrets

Any shared secret used for RADIUS/UDP or RADIUS/TLS MUST NOT be used for TLS-PSK.

It is RECOMMENDED that RADIUS clients and server track all used shared secrets and PSKs, and then verify that the following requirements all hold true:

- *no shared secret is used for more than one RADIUS client
- *no PSK is used for more than one RADIUS client
- *no shared secret is used as a PSK
- *no PSK is used as a shared secret

There may be use-cases for using one shared secret across multiple RADIUS clients. There may similarly be use-cases for sharing a PSK across multiple RADIUS clients. Details of the possible attacks on reused PSKs are given in [RFC9257] Section 4.1.

There are few, if any, use-cases for using a PSK as a shared secret, or vice-versa.

Implementaions MUST NOT provide user interfaces which allow both PSKs and shared secrets to be entered at the same time. Only one or the other must be present. Implementations MUST NOT use a "shared secret" field as a way for administrators to enter PSKs. The PSK entry fields MUST be labelled as being related to PSKs, and not to shared secrets.

4.2. PSK Identities

It is RECOMMENDED that systems follow the directions of [RFC9257] Section 6.1.1 for the use of external PSK identies in TLS. Note that the PSK identity is sent in the clear, and is therefore visible to attackers. Where privacy is desired, the PSK identity could be either an opaque token generated cryptographically, or perhaps in the form of a Network Access Identifier (NAI) [RFC7542], where the "user" portion is an opaque token. For example, an NAI could be "68092112@example.com". If the attacker already knows that the client is associated with "example.com", then using that domain name in the PSK identity offers no additional information. In constrast, the "user" portion needs to be both unique to the client and private, so using an opaque token there is a more secure approach.

Implementations MUST support PSK identies of 128 octets, and SHOULD support longer PSK identities. We note that while TLS provides for PSK identities of up to 2^16-1 octets in length, there are few practical uses for extremely long PSK identities.

4.3. PSK and PSK Identity Sharing

While administrators may desire to share PSKs and/or PSK identities across multiple systems, such usage is NOT RECOMMENDED. Details of

the possible attacks on reused PSKs are given in [RFC9257] Section 4.1.

Implementations MUST support configuring a unique PSK and PSK identity for each possible client-server relationship. This configuration allows administrators desiring security to use unique PSKs for each such relationship. This configuration also allows administrators to re-use PSKs and PSK identies where local policies permit.

Implementations SHOULD warn administrators if the same PSK identity and/or PSK is used for multiple client-server relationships.

5. Guidance for RADIUS clients

TLS uses certificates in most common uses. However, we recognize that it may be difficult to fully upgrade client implementations to allow for certificates to be used with RADIUS/TLS and RADIUS/DTLS. Client implementations therefore MUST allow the use of a pre-shared key (TLS-PSK). The client implementation can then expose a flag "TLS yes / no", and then fields which ask for the PSK identity and PSK itself.

Implementations MUST use ECDH cipher suites. Implementations MUST implement the recommended cipher suites in [RFC9325] Section 4.2 for TLS 1.2, and in [RFC9325] Section 4.2 for TLS 1.3.

5.1. PSK Identities

[RFC6614] is silent on the subject of PSK identities, which is an issue that we correct here. Guidance is required on the use of PSK identities, as the need to manage identities associated with PSK is a new requirement for NAS management interfaces, and is a new requirement for RADIUS servers.

RADIUS systems implementing TLS-PSK MUST support identities as per [RFC4279] Section 5.3, and MUST enable configuring TLS-PSK identities in management interfaces as per [RFC4279] Section 5.4.

RADIUS shared secrets cannot safely be used as TLS-PSKs. To prevent confusion between shared secrets and TLS-PSKs, management interfaces and APIs need to label PSK fields as "PSK" or "TLS-PSK", rather than "shared secret

Where dynamic server lookups [RFC7585] are not used, RADIUS clients MUST still permit the configuration of a RADIUS server IP address.

6. Guidance for RADIUS Servers

The following section(s) describe guidance for RADIUS server implementations and deployments.

6.1. Identifying and filtering clients

RADIUS/UDP and RADIUS/TCP identify clients by source IP address. This practice is no longer needed when TLS transport is used, as the client can instead be identified via TLS information such as PSK identity, client certificate, etc.

When a RADIUS server implements TLS-PSK, it MUST use the PSK identity as the logical identifier for a RADIUS client instead of the IP address as was done with RADIUS/UDP. That is, instead of associating a source IP address with a shared secret, the RADIUS server instead associates a PSK identity with a pre-shared key. In effect, the PSK identity replaces the source IP address of the connection as the client identifier.

For example, when a RADIUS server receives a RADIUS/UDP packet, it normally looks up the source IP address, finds a client definition, and that client definition contains a shared secret. The packet is then authenticated (or not) using that shared secret.

When TLS-PSK is used, the RADIUS server instead receives a TLS connection request which contains a PSK identity. That identity is then used to find a client definition, and that client definition contains a PSK. The TLS connection is then authenticated (or not) using that PSK.

Each RADIUS client MUST be configured with a unique PSK, which implies a unique PSK identifier for each RADIUS client. To enforce the use of unique PSKs, RADIUS servers accepting TLS-PSK MUST require that a PSK identifier and PSK can be associated with each RADIUS client.

RADIUS servers MUST be able to look up PSK identity in a subsystem which then returns the actual PSK.

RADIUS servers MUST support IP address and network filtering of the source IP address for all TLS connections. In many situations a RADIUS server does not need to allow connections from the entire Internet. As such, it can increase security to limit permitted connections to a small list of networks.

For example, a RADIUS server be configured to be accept connections from a source network of 192.0.2/24. The RADIUS server could therefore discard any TLS connection request which comes from a source IP address outside of that network. In that case, there is no need to examine the PSK identity or to find the client definition. Instead, the IP source filtering policy would deny the connection before any TLS communication had been performed.

RADIUS servers SHOULD be able to limit certain PSK identifiers to certain network ranges or IP addresses. This filtering can catch configuration errors. That is, if a NAS is known to have a dynamic IP address within a particular subnet, the server should limit use of the NASes PSK to that subnet.

For example, as with the example above, the RADIUS server be configured to be accept connections from a source network of 192.0.2/24. The RADIUS server may be configured to with a PSK idrnity "system1", and then also configured to associate that PSK identity with the source IP address 192.0.2.16. In that case, if the server receives a connection request from the source IP address 192.0.2.16 with PSK identity other than "system1", then the connection could be rejected. Similarly, if the server receives a connection request from the source IP address other than 192.0.2.16 but which uses the PSK identity "system1", then the connection could also be rejected.

The use of PSK identities as client identifiers does not prevent RADIUS servers from performing source IP filtering of incoming packets or connections. Instead, the use of PSK identities as client identifiers means that source IP addresses are no longer required to be associated with RADIUS clients.

Note that as some clients may have dynamic IP addresses, it is possible for a one PSK identity to appear at different source IP addresses over time. In addition, as there may be many clients behind one NAT gateway, there may be multiple RADIUS clients using one public IP address. RADIUS servers MUST support multiple PSKs at one source IP address, and MUST support a unique PSK identity for each unique client which is deployed in such a scenario.

In those use-cases, the RADIUS server should either not use source IP address filtering, or should apply source IP filtering rules which permit those use-cases. This filtering must therefore be flexible to allow all of the above behaviors, and be configurable by administrators to match their needs.

RADIUS servers SHOULD tie PSK identities to a particular permitted IP address or permitted network, as doing so will lower the risk if a PSK is leaked. RADIUS servers MUST permit multiple clients to share one permitted IP address or network.

7. Privacy Considerations

We make no changes over [RFC6614] and [RFC7360].

8. Security Considerations

The primary focus of this document is addressing security considerations for RADIUS.

9. IANA Considerations

There are no IANA considerations in this document.

RFC Editor: This section may be removed before final publication.

10. Acknowledgements

TBD.

11. Changelog

*00 - initial version

*01 - update examples

12. References

12.1. Normative References

[BCP14] 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>.

[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>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC4279] Eronen, P., Ed. and H. Tschofenig, Ed., "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, DOI 10.17487/RFC4279, December 2005, <https://www.rfc-editor.org/info/rfc4279>.

[RFC7585] Winter, S. and M. McCauley, "Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS Based on the Network Access

Identifier (NAI)", RFC 7585, DOI 10.17487/RFC7585, October 2015, <https://www.rfc-editor.org/info/rfc7585>.

[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>.

[RFC9258] Benjamin, D. and C. A. Wood, "Importing External Pre-Shared Keys (PSKs) for TLS 1.3", RFC 9258, DOI 10.17487/RFC9258, July 2022, <https://www.rfc-editor.org/info/rfc9258>.

12.2. Informative References

[RFC6613] DeKok, A., "RADIUS over TCP", RFC 6613, DOI 10.17487/RFC6613, May 2012, <https://www.rfc-editor.org/info/rfc6613>.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <https://www.rfc-editor.org/info/rfc6614>.

[RFC7360] DeKok, A., "Datagram Transport Layer Security (DTLS) as a Transport Layer for RADIUS", RFC 7360, DOI 10.17487/RFC7360, September 2014, <https://www.rfc-editor.org/info/rfc7360>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/info/rfc7542>.

[RFC8937] Cremers, C., Garratt, L., Smyshlyaev, S., Sullivan, N., and C. Wood, "Randomness Improvements for Security Protocols", RFC 8937, DOI 10.17487/RFC8937, October 2020, <https://www.rfc-editor.org/info/rfc8937>.

[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>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security

(DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

Author's Address

Alan DeKok
FreeRADIUS

Email: aland@freeradius.org