Internet Engineering Task Force (IETF)
Request for Comments: 5929
Category: Standards Track
ISSN: 2070-1721

J. Altman
Secure Endpoints
N. Williams
Oracle
L. Zhu
Microsoft Corporation

July 2010

Channel Bindings for TLS

Abstract

This document defines three channel binding types for Transport Layer Security (TLS), tls-unique, tls-server-end-point, and tls-unique-for-telnet, in accordance with RFC 5056 (On Channel Binding).

Note that based on implementation experience, this document changes the original definition of 'tls-unique' channel binding type in the channel binding type IANA registry.

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 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5929.

Table of Contents

1. Introduction

Subsequent to the publication of "On Channel Bindings" [RFC5056], three channel binding types for Transport Layer Security (TLS) were proposed, reviewed, and added to the IANA channel binding type registry, all in accordance with [RFC5056]. Those channel binding types are: 'tls-unique', 'tls-server-end-point', and 'tls-unique-for-telnet'. It has become desirable to have these channel binding types re-registered through an RFC so as to make it easier to reference them, and to correct them to describe actual implementations. This document does just that. The authors of those three channel binding types have transferred, or have indicated that they will transfer, "ownership" of those channel binding types to the IESG.

We also provide some advice on the applicability of these channel binding types, as well as advice on when to use which. Additionally, we provide an abstract API that TLS implementors should provide, by which to obtain channel bindings data for a TLS connection.

WARNING: it turns out that the first implementor implemented and deployed something rather different than what was described in the IANA registration for 'tls-unique'. Subsequently, it was decided that we should adopt that form of 'tls-unique'. This means that this document makes a backwards-incompatible change to 'tls-unique'. See Section 8 for more details.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. The 'tls-unique' Channel Binding Type

IANA updated the registration of the 'tls-unique' channel binding type to match the description below. There are material and substantial changes from the original registration, both in the description as well as registration meta-data (such as registration ownership).

3.1. Description

Description: The first TLS Finished message sent (note: the Finished struct, not the TLS record layer message containing it) in the most recent TLS handshake of the TLS connection being bound to (note: TLS connection, not session, so that the channel binding is specific to each connection regardless of whether session resumption is used). If TLS renegotiation takes place before the channel binding

operation, then the first TLS Finished message sent of the latest/ inner-most TLS connection is used. Note that for full TLS handshakes, the first Finished message is sent by the client, while for abbreviated TLS handshakes (session resumption), the first Finished message is sent by the server.

WARNING: The definition, security, and interoperability considerations of this channel binding type have changed since the original registration. Implementors should read the document that last updated this registration for more information.

Interoperability note:

This definition of 'tls-unique' means that a channel's bindings data may change over time, which in turn creates a synchronization problem should the channel's bindings data change between the time that the client initiates authentication with channel binding and the time that the server begins to process the client's first authentication message. If that happens, the authentication attempt will fail spuriously.

Based on the fact that while servers may request TLS renegotiation, only clients may initiate it, this synchronization problem can be avoided by clients and servers as follows: server applications MUST NOT request TLS renegotiation during phases of the application protocol during which application-layer authentication occurs. Client applications SHOULD NOT initiate TLS renegotiation between the start and completion of authentication.

The rationale for making the server behavior a requirement while the client behavior is only a recommendation is that there typically exist TLS APIs for requesting renegotiation on the server side of a TLS connection, while many client TLS stacks do not provide fine-grained control over when TLS renegotiation occurs.

Application protocols SHOULD be designed in such a way that a server would never need to request TLS renegotiation immediately before or during application-layer authentication.

3.2. Registration

- Channel binding unique prefix: tls-unique
- Channel binding type: unique
- Channel type: TLS [RFC5246]

- Published specification: <RFC 5929>
- Channel binding is secret: no
- Description: <See specification>
- Intended usage: COMMON
- Person and email address to contact for further information: Larry Zhu (larry.zhu@microsoft.com), Nicolas Williams (Nicolas.Williams@oracle.com).
- Owner/Change controller name and email address: IESG.
- Expert reviewer name and contact information: IETF TLS WG (tls@ietf.org, failing that, ietf@ietf.org)
- Note: see the published specification for advice on the applicability of this channel binding type.

4. The 'tls-server-end-point' Channel Binding Type

IANA updated the registration of the 'tls-server-end-point' channel binding type to match the description below. Note that the only material changes from the original registration are: the "owner" (now the IESG), the contacts, the published specification, and a note indicating that the published specification should be consulted for applicability advice. References were added to the description. All other fields of the registration are copied here for the convenience of readers.

4.1. Description

Description: The hash of the TLS server's certificate [RFC5280] as it appears, octet for octet, in the server's Certificate message. Note that the Certificate message contains a certificate_list, in which the first element is the server's certificate.

The hash function is to be selected as follows:

- if the certificate's signatureAlgorithm uses a single hash function, and that hash function is either MD5 [RFC1321] or SHA-1 [RFC3174], then use SHA-256 [FIPS-180-3];
- if the certificate's signatureAlgorithm uses a single hash function and that hash function neither MD5 nor SHA-1, then use the hash function associated with the certificate's signatureAlgorithm;

- if the certificate's signatureAlgorithm uses no hash functions or uses multiple hash functions, then this channel binding type's channel bindings are undefined at this time (updates to is channel binding type may occur to address this issue if it ever arises).

The reason for using a hash of the certificate is that some implementations need to track the channel binding of a TLS session in kernel-mode memory, which is often at a premium.

4.2. Registration

- Channel binding unique prefix: tls-server-end-point
- Channel binding type: end-point
- Channel type: TLS [RFC5246]
- Published specification: <RFC 5929>
- Channel binding is secret: no
- Description: <See specification>
- Intended usage: COMMON
- Person and email address to contact for further information: Larry Zhu (larry.zhu@microsoft.com), Nicolas Williams (Nicolas.Williams@oracle.com).
- Owner/Change controller name and email address: IESG.
- Expert reviewer name and contact information: IETF TLS WG (tls@ietf.org, failing that, ietf@ietf.org)
- Note: see the published specification for advice on the applicability of this channel binding type.

5. The 'tls-unique-for-telnet' Channel Binding Type

IANA updated the registration of the 'tls-unique-for-telnet' channel binding type to match the description below. Note that the only material changes from the original registration are: the "owner" (now the IESG), the contacts, the published specification, and a note indicating that the published specification should be consulted for applicability advice. The description is also clarified. We also moved the security considerations notes to the security considerations section of this document. All other fields of the registration are copied here for the convenience of readers.

5.1. Description

Description: There is a proposal for adding a "StartTLS" extension to TELNET, and a channel binding extension for the various TELNET AUTH mechanisms whereby each side sends the other a "checksum" (MAC -- message authentication code) of their view of the channel's bindings. The client uses the TLS Finished messages (note: the Finished struct) sent by the client and server, each concatenated in that order and in their clear text form, of the first TLS handshake to which the connection is being bound. The server does the same but in the opposite concatenation order (server, then client).

5.2. Registration

- Channel binding unique prefix: tls-unique-for-telnet
- Channel binding type: unique
- Channel type: TLS [RFC5246]
- Published specification: <RFC 5929>
- Channel binding is secret: no
- Description: <See specification>
- Intended usage: COMMON
- Person and email address to contact for further information: Jeff Altman (jaltman@secure-endpoints.com), Nicolas Williams (Nicolas.Williams@oracle.com).
- Owner/Change controller name and email address: IESG.
- Expert reviewer name and contact information: IETF TLS WG (tls@ietf.org, failing that, ietf@ietf.org)
- Note: see the published specification for advice on the applicability of this channel binding type.

6. Applicability of TLS Channel Binding Types

The 'tls-unique-for-telnet' channel binding type is only applicable to TELNET [RFC0854] and is available for all TLS connections.

The 'tls-unique' channel binding type is available for all TLS connections, while 'tls-server-end-point' is only available when TLS cipher suites with server certificates are used, specifically: cipher

suites that use the Certificate handshake message, which typically involve the use of PKIX [RFC5280]. For example, 'tls-server-end-point' is available when using TLS ciphers suites such as (this is not an exhaustive list):

- TLS_DHE_DSS_WITH_*
- TLS_DHE_RSA_WITH_*
- TLS_DH_DSS_WITH_*
- TLS_DH_RSA_WITH_*
- TLS_ECDHE_ECDSA_WITH_*
- TLS_ECDHE_RSA_WITH_*
- TLS_ECDH_ECDSA_WITH_*
- TLS_ECDH_RSA_WITH_*
- TLS_RSA_PSK_WITH_*
- TLS_RSA_WITH_*
- TLS_SRP_SHA_DSS_WITH_*
- TLS_SRP_SHA_RSA_WITH_*

but is not available when using TLS cipher suites such as (this is not an exhaustive list):

- TLS_DHE_PSK_WITH_*
- TLS_DH_anon_WITH_*
- TLS_ECDHE_PSK_WITH_*
- TLS_ECDH_anon_WITH_*
- TLS_KRB5_WITH_*
- TLS_PSK_WITH_*
- TLS_SRP_SHA_WITH_*

'tls-server-end-point' is also not applicable for use with OpenPGP server certificates [RFC5081] [RFC4880] (since these don't use the Certificate handshake message).

Therefore, 'tls-unique' is applicable to more contexts than 'tls-server-end-point'. However, 'tls-server-end-point' may be used with existing TLS server-side proxies ("concentrators") without modification to the proxies, whereas 'tls-unique' may require firmware or software updates to server-side proxies. Therefore there may be cases where 'tls-server-end-point' may interoperate but where 'tls-unique' may not.

Also, authentication mechanisms may arise that depend on channel bindings to contribute entropy, in which case unique channel bindings would always have to be used in preference to end-point channel bindings. At this time there are no such mechanisms, though one such SASL mechanism has been proposed. Whether such mechanisms should be allowed is out of scope for this document.

For many applications, there may be two or more potentially applicable TLS channel binding types. Existing security frameworks (such as the GSS-API [RFC2743] or the SASL [RFC4422] GS2 framework [RFC5801]) and security mechanisms generally do not support negotiation of channel binding types. Therefore, application peers need to agree a priori as to what channel binding type to use (or agree to rules for deciding what channel binding type to use).

The specifics of whether and how to negotiate channel binding types are beyond the scope of this document. However, it is RECOMMENDED that application protocols making use of TLS channel bindings, use 'tls-unique' exclusively, except, perhaps, where server-side proxies are common in deployments of an application protocol. In the latter case an application protocol MAY specify that 'tls-server-end-point' channel bindings must be used when available, with 'tls-unique' being used when 'tls-server-end-point' channel bindings are not available. Alternatively, the application may negotiate which channel binding type to use, or may make the choice of channel binding type configurable.

Specifically, application protocol specifications MUST indicate at least one mandatory to implement channel binding type, MAY specify a negotiation protocol, MAY allow for out-of-band negotiation or configuration, and SHOULD have a preference for 'tls-unique' over 'tls-server-end-point'.

7. Required Application Programming Interfaces

TLS implementations supporting the use of 'tls-unique' and/or 'tls-unique-for-telnet' channel binding types MUST provide application programming interfaces by which applications (clients and servers both) may obtain the channel bindings for a TLS connection. Such interfaces may be expressed in terms of extracting the channel bindings data for a given connection and channel binding type. Alternatively, the implementor may provide interfaces by which to obtain the initial client Finished message, the initial server Finished message, and/or the server certificate (in a form that matches the description of the 'tls-server-end-point' channel binding type). In the latter case, the application has to have knowledge of the channel binding type descriptions from this document. This document takes no position on which form these application programming interfaces must take.

TLS implementations supporting TLS renegotiation SHOULD provide APIs that allow applications to control when renegotiation can take place. For example, a TLS client implementation may provide a "callback" interface to indicate that the server requested renegotiation, but may not start renegotiation until the application calls a function to indicate that now is a good time to renegotiate.

8. Description of Backwards-Incompatible Changes Made Herein to 'tls-unique'

The original description of 'tls-unique' read as follows:

> |OLD| Description: The client's TLS Finished message (note: the
> |OLD| Finished struct) from the first handshake of the connection
> |OLD| (note: connection, not session, so that the channel binding
> |OLD| is specific to each connection regardless of whether session
> |OLD| resumption is used).

Original 'tls-unique' description

In other words: the client's Finished message from the first handshake of a connection, regardless of whether that handshake was a full or abbreviated handshake, and regardless of how many subsequent handshakes (renegotiations) might have followed.

As explained in Section 1, this is no longer the description of 'tls-unique', and the new description is not backwards compatible with the original except in the case of TLS connections where: a) only one handshake has taken place before application-layer authentication, and b) that one handshake was a full handshake.

This change has a number of implications:

- Backwards-incompatibility. It is possible that some implementations of the original 'tls-unique' channel binding type have been deployed. We know of at least one TLS implementation that exports 'tls-unique' channel bindings with the original semantics, but we know of no deployed application using the same. Implementations of the original and new 'tls-unique' channel binding type will only interoperate when: a) full TLS handshakes are used, and b) TLS renegotiation is not used.

- Security considerations -- see Section 10.

- Interoperability considerations. As described in Section 3, the new definition of the 'tls-unique' channel binding type has an interoperability problem that may result in spurious authentication failures unless the application implements one or both of the techniques described in that section.

9. IANA Considerations

IANA updated three existing channel binding type registrations. See the rest of this document.

10. Security Considerations

The Security Considerations sections of [RFC5056], [RFC5246], and [RFC5746] apply to this document.

The TLS Finished messages (see Section 7.4.9 of [RFC5246]) are known to both endpoints of a TLS connection and are cryptographically bound to it. For implementations of TLS that correctly handle renegotiation [RFC5746], each handshake on a TLS connection is bound to the preceding handshake, if any. Therefore, the TLS Finished messages can be safely used as a channel binding provided that the authentication mechanism doing the channel binding conforms to the requirements in [RFC5056]. Applications utilizing 'tls-unique' channel binding with TLS implementations without support for secure renegotiation [RFC5746] MUST ensure that ChangeCipherSpec has been used in any and all renegotiations prior to application-layer authentication, and MUST discard any knowledge learned from the server prior to the completion of application-layer authentication.

The server certificate, when present, is also cryptographically bound to the TLS connection through its use in key transport and/or authentication of the server (either by dint of its use in key transport, by its use in signing key agreement, or by its use in key

agreement). Therefore, the server certificate is suitable as an end-point channel binding as described in [RFC5056].

10.1. Cryptographic Algorithm Agility

The 'tls-unique' and 'tls-unique-for-telnet' channel binding types do not add any use of cryptography beyond that used by TLS itself. Therefore, these two channel binding types add no considerations with respect to cryptographic algorithm agility.

The 'tls-server-end-point' channel binding type consists of a hash of a server certificate. The reason for this is to produce manageably small channel binding data, as some implementations will be using kernel-mode memory (which is typically scarce) to store these. This use of a hash algorithm is above and beyond TLS's use of cryptography, therefore the 'tls-server-end-point' channel binding type has a security consideration with respect to hash algorithm agility. The algorithm to be used, however, is derived from the server certificate's signature algorithm as described in Section 4.1; to recap: use SHA-256 if the certificate signature algorithm uses MD5 or SHA-1, else use whatever hash function the certificate uses (unless the signature algorithm uses no hash functions or more than one hash function, in which case 'tls-server-end-point' is undefined). The construction of 'tls-server-end-point' channel bindings is not directly hash-agile (since no negotiation of hash function is provided for), but it is hash-agile nonetheless. The hash agility of 'tls-server-end-point' channel bindings derives from PKIX and TLS.

Current proposals for randomized signatures algorithms [RHASH] [NIST-SP.800-106.2009] use hash functions in their construction -- a single hash function in each algorithm. Therefore, the 'tls-server-end-point' channel binding type should be available even in cases where new signatures algorithms are used that are based on current randomized hashing proposals (but we cannot guarantee this, of course).

10.2. On Disclosure of Channel Bindings Data by Authentication Mechanisms

When these channel binding types were first considered, one issue that some commenters were concerned about was the possible impact on the security of the TLS channel, of disclosure of the channel bindings data by authentication mechanisms. This can happen, for example, when an authentication mechanism transports the channel bindings data, with no confidentiality protection, over other transports (for example, in communicating with a trusted third party), or when the TLS channel provides no confidentiality

protection and the authentication mechanism does not protect the confidentiality of the channel bindings data. This section considers that concern.

When the TLS connection uses a cipher suite that does not provide confidentiality protection, the TLS Finished messages will be visible to eavesdroppers, regardless of what the authentication mechanism does. The same is true of the server certificate which, in any case, is generally visible to eavesdroppers. Therefore we must consider our choices of TLS channel bindings here to be safe to disclose by definition -- if that were not the case, then TLS with cipher suites that don't provide confidentiality protection would be unsafe. Furthermore, the TLS Finished message construction depends on the security of the TLS PRF, which in turn needs to be resistant to key recovery attacks, and we think that it is, as it is based on HMAC, and the master secret is, well, secret (and the result of key exchange).

Note too that in the case of an attempted active man-in-the-middle attack, the attacker will already possess knowledge of the TLS Finished messages for both inbound and outbound TLS channels (which will differ, given that the attacker cannot force them to be the same). No additional information is obtained by the attacker from the authentication mechanism's disclosure of channel bindings data -- the attacker already has it, even when cipher suites providing confidentiality protection are provided.

None of the channel binding types defined herein produce channel bindings data that must be kept secret. Moreover, none of the channel binding types defined herein can be expected to be private (known only to the end-points of the channel), except that the unique TLS channel binding types can be expected to be private when a cipher suite that provides confidentiality protection is used to protect the Finished message exchanges and the application data records containing application-layer authentication messages.

11. References

11.1. Normative References

[FIPS-180-3] United States of America, National Institute of Standards and Technology, "Secure Hash Standard", Federal Information Processing Standard (FIPS) 180-3, October 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5056] Williams, N., "On the Use of Channel Bindings to Secure Channels", RFC 5056, November 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension", RFC 5746, February 2010.

11.2. Informative References

[NIST-SP.800-106.2009] National Institute of Standards and Technology, "NIST Special Publication 800-106: Randomized Hashing for Digital Signatures", February 2009.

[RFC0854] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, May 1983.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[RFC3174] Eastlake, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

[RFC4422] Melnikov, A., Ed., and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[RFC5081] Mavrogiannopoulos, N., "Using OpenPGP Keys for Transport Layer Security (TLS) Authentication", RFC 5081, November 2007.

[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, May 2008.

[RFC5801] Josefsson, S. and N. Williams, "Using Generic Security Service Application Program Interface (GSS-API) Mechanisms in Simple Authentication and Security Layer (SASL): The GS2 Mechanism Family", RFC 5801, July 2010.

[RHASH] Halevi, S. and H. Krawczyk, "Strengthening Digital Signatures via Randomized Hashing", Work in Progress, October 2007.

Authors' Addresses

Jeff Altman
Secure Endpoints
255 W 94TH ST PHB
New York, NY 10025
US

EMail: jaltman@secure-endpoints.com

Nicolas Williams
Oracle
5300 Riata Trace Ct
Austin, TX 78727
US

EMail: Nicolas.Williams@oracle.com

Larry Zhu
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

EMail: larry.zhu@microsoft.com