Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2011

P. Hoffman
VPN Consortium

July 8, 2010

Additional Master Secret Inputs for TLS
draft-hoffman-tls-master-secret-input-02

Abstract

This document describes a mechanism for using additional master secret inputs with Transport Layer Security (TLS) and Datagram TLS (DTLS).

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 http://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 January 9, 2011.

Copyright Notice

1. Introduction

Some TLS 1.2[RFC5246] and DTLS 1.2 [4347bis] extensions want to mix particular data into the calculation of the master_secret. This mixing creates a cryptographic binding of the added material directly into the secret that is used to protect the TLS session. For example, some systems want to be sure that there is sufficient randomness in the TLS master_secret, and this can be accomplished by adding it directly to the master_secret calculations.

This document describes a framework for TLS and DTLS extensions to meet these requirements. In an extension that uses this framework, a client and server provide data in the handshake using normal TLS extensions, and then this data is combined with the ClientHello and ServerHello random values during the derivation of the master_secret.

Extensions that specify data to be added to the master secret are called "extensions with master secret input". An extension with master secret input must specify the additional input that comes from the client and/or the server. Note that the term "and/or" is used here because the definition of the extension might cause input to the master secret to come from only one of the participants.

Note that extensions that do not specify that they are extensions with master secret input cannot be extensions with master secret input. That is, every extension that does not call itself an extension with master secret input is treated just like a normal extension. Also note that this document only describes a framework; if an extension uses this framework, and a client and server both implement the extension, no signaling about the use of master secret input is needed: that comes as part of the extension definition itself.

Use of one or more of these extensions changes the way that the master secret is calculated in TLS and DTLS. That is, if the handshake has no extensions, or only extensions that are not

extensions with master secret input, the master secret calculation is unchanged.

1.1. 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].

2. Master Secret Calculation Modifications for TLS and DTLS

When an extension with master secret input is present in the handshake, the additional master secret input values MUST be mixed into the pseudorandom function (PRF) calculation along with the client and server random values during the computation of the master_secret. For the calculation of the master secret, the extensions MUST be sorted by extension type order. Note that TLS 1.2 specifies that there can only be one extension per type, and the extensions can appear in mixed order.

Each extension with master secret input adds its own specified input, called "additional_ms_input_1" for the extension with master secret input that has the lowest type number, "additional_ms_input_2" for the extension with master secret input with the second lowest type number, and so on.

The calculation of the master_secret becomes:

```
master_secret = PRF(pre_master_secret, "master secret",
                    ClientHello.random +
                    ClientHello.additional_ms_input_1 +
                    ClientHello.additional_ms_input_2 +
                    . . .
                    ClientHello.additional_ms_input_N +
                    ServerHello.random +
                    ServerHello.additional_ms_input_1 +
                    ServerHello.additional_ms_input_2 +
                    . . .
                    ServerHello.additional_ms_input_N +
                    )[0..47];
```

Using the specified order of the additional_ms_input_n fields in the master_secret is required for interoperability. Otherwise, a server and a client would not know how to unambiguously calculate the same master_secret.

3. Security Considerations

This modification to TLS and DTLS increases the amount of data that an attacker can inject into the master secret calculation. This potentially would allow an attacker who had partially compromised the inputs to the master secret calculation greater scope for influencing the output. Hash-based PRFs like the one used in TLS master secret calculations are designed to be fairly indifferent to the input size.

The additional master secret input may have no entropy; in fact, it might be completely predictable to an attacker. TLS is designed to function correctly even when the PRF used in the master secret calculation has a great deal of predictable material because the PRF is used to generate distinct keying material for each connection. Thus, even in the face of completely predictable additional master secret input values, no harm is done to the resulting PRF output. When there is entropy in these values, that entropy is reflected in the PRF output.

4. IANA Considerations

[[This section should be removed at the time of RFC publication.]] No IANA registries are changed or created by this document. At the time that this document was written, none of the extensions in the IANA TLS registry (http://www.iana.org/assignments/tls-extensiontype-values/ tls-extensiontype-values.xhtml) are extensions with master secret input.

5. Acknowledgements

Much of the text in this document is derived from text written by Eric Rescorla, Margaret Salter, and Jerry Solinas.

6. Normative References

[4347bis] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security version 1.2", draft-ietf-tls-rfc4347-bis (work in progress), October 2009.

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

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

Author's Address

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org