Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 31, 2008

E. Rescorla
RTFM, Inc.
M. Salter
National Security Agency

April 29, 2008

Extended Random Values for TLS
draft-rescorla-tls-extended-random-00.txt

Status of this Memo

Copyright Notice

Abstract

This document describes an extension for using larger client and server Random values with Transport Layer Security (TLS) and Datagram TLS (DTLS).

Table of Contents

1. Introduction

TLS [I-D.ietf-tls-rfc4346-bis] and DTLS [RFC4347] use a 32-byte "Random" value consisting of a 32-bit time value time and 28 randomly generated bytes:

```
      struct {
         uint32 gmt_unix_time;
         opaque random_bytes[28];
      } Random;
```

The client and server each contribute a Random value which is then mixed with secret keying material to produce the final per-association keying material.

The United States Department of Defense has requested a TLS mode which allows the use of longer public randomness values for use with high security level cipher suites like those specified in Suite B [I-D.rescorla-tls-suiteb]. The rationale for this as stated by DoD is that the public randomness for each side should be at least twice as long as the security level for cryptographic parity, which makes the 224 bits of randomness provided by the current TLS random values insufficient.

This document specifies an extension which allows for additional randomness to be exchanged in the Hello messages.

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 ExtendedRandom Extension

This document defines a new TLS extension called "extended_random".

The "extended_random" extension carried in a new TLS extension called "ExtendedRandom".

```
     struct {
         opaque extended_random_value<0..2^16-1>;
     } ExtendedRandom;
```

The extended_random_value MUST be a randomly generated byte string. A cryptographically secure PRNG [RFC4086] SHOULD be used.

3.1. Negotiating the ExtendedRandom Extension

The client requests support for the extended randomness feature by sending an "extended_random" extension in its ClientHello. The "extension_data" field contains an ExtendedRandom value.

When a server which does not recognize the "extended_random" extension receives one, it will ignore it as required. A server which recognizes the extension MAY choose to ignore it, in which case it SHOULD continue with the exchange as if it had not received the extension.

If the server wishes to use the extended randomness feature, it MUST send its own "extended_random" extension with an extended_random_value equal in length to the client's extended_random_value. Clients SHOULD check the length of the server's extended_random_value and generate a fatal "illegal_parameter" error if it is present but does does not match the length that was transmitted in the ClientHello.

Because TLS does not permit servers to request extensions which the client did not offer, the client may not offer the "extended_random" extension even if the server requires it. In this case, the server should generate a fatal "handshake_failure" alert.

Because there is no way to mark extensions as critical, the server may ignore the "extended_random" extension even though the client requires it. If a client requires the extended randomness input feature but the server does not negotiate it, the client SHOULD generate a fatal "handshake_failure" alert.

3.2. PRF Modifications

When the extended randomness feature is in use, the extended random values MUST be mixed into the PRF along with the client and server random values during the PMS->MS conversion. Thus, the PRF becomes:

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

Because new extensions may not be introduced in resumed handshakes, mixing in the extended inputs during the MS->keying material conversion would simply involve mixing in the same material twice. Therefore, the extended random inputs are only used when the PMS is converted into the MS.

4. Security Considerations

4.1. Threats to TLS

When this extension is in use it increases the amount of data that an attacker can inject into the PRF. This potentially would allow an attacker who had partially compromised the PRF greater scope for influencing the output. Hash-based PRFs like the one in TLS are designed to be fairly indifferent to the input size (the input is already greater than the block size of most hash functions), however there is currently no proof that a larger input space would not make attacks easier.

Another concern is that bad implementations might generate low entropy extented random values. TLS is designed to function correctly even when fed low-entropy random values because they are primarily used to generate distinct keying material for each connection.

4.2. Scope of Randomness

TLS specifies that when a session is resumed the extensions from the original connection are used:

> If, on the other hand, the older session is resumed, then the server MUST ignore the extensions and send a server hello containing none of the extension types. In this case, the functionality of these extensions negotiated during the original session initiation is applied to the resumed session.

This motivates why the the extended randomness does not get mixed into the PRF when generating the keying material from the master secret. Because the same values would be used for every connection in a session, they would not provide any differentiation in the keying material between the connections.

5. IANA Considerations

This document defines an extension to TLS, in accordance with [I-D.ietf-tls-rfc4366-bis]:

```
  enum { extended_random (??) } ExtensionType;
```

[[NOTE: These values need to be assigned by IANA]]

6. Acknowledgements

This work was supported by the US Department of Defense.

7. References

7.1. Normative References

[I-D.ietf-tls-rfc4346-bis]
Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", draft-ietf-tls-rfc4346-bis-10 (work in progress), March 2008.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

7.2. Informative References

[I-D.ietf-tls-rfc4366-bis]
3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", draft-ietf-tls-rfc4366-bis-02 (work in progress), February 2008.

[I-D.rescorla-tls-suiteb]
Salter, M. and E. Rescorla, "Suite B Cipher Suites for TLS", draft-rescorla-tls-suiteb-02 (work in progress), April 2008.

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

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

Authors' Addresses

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Email: ekr@rtfm.com

Margaret Salter
National Security Agency
9800 Savage Rd.
Fort Meade 20755-6709
USA

Email: msalter@restarea.ncsc.mil

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).