RTCWEB
Internet-Draft
Intended status: Standards Track
Expires: May 24, 2014

M. Thomson
Mozilla
D. Wing
C. Jennings
Cisco

November 20, 2013

Gaining and Maintaining Consent for Real-Time Applications
draft-thomson-rtcweb-consent-00

Abstract

This document describes how DTLS provides a WebRTC application a clear indication that a receiver is willing to receive packets. Mechanisms are described for maintaining that consent are described.

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 May 24, 2014.

Table of Contents

1. Introduction

In addition to establishing connectivity, Interactive Connectivity Establishment (ICE) [RFC5245] has been used in real-time applications to establish that a peer is willing to receive packets.

This document describes how Datagram Transport Layer Security (DTLS) [RFC6347] is sufficient for establishing consent to receive packets, plus how this consent can be continuously refreshed.

This also uses Application-Layer Protocol Negotiation (ALPN) [I-D.ietf-tls-applayerprotoneg] to restrict that consent to specific uses. Application protocol tokens are defined for the Real-Time Protocol (RTP) [RFC3550] over DTLS-SRTP [RFC5764], WebRTC data channels [I-D.ietf-rtcweb-data-channel] and a multiplexed combination of these two protocols.

1.1. Conventions and Terminology

At times, this document falls back on shorthands for establishing interoperability requirements on implementations: the capitalized words "MUST", "SHOULD" and "MAY". These terms are defined in [RFC2119].

2. Obtaining and Maintaining Receive Consent

An endpoint MUST NOT send application data (in WebRTC, RTP or SCTP data) on a DTLS connection unless the receiving endpoint consents to receive the data.

An endpoint that initiates or responds to a DTLS handshake that negotiates a specific application layer protocol (see Section 3) explicitly consents to receive packets containing the described protocol.

Consent expires after a fixed amount of time. Explicit consent to receive is indicated by the receiving endpoint sending authenticated packets from the inverted 5-tuple. An endpoint uses the receipt of packets as an indication that the remote endpoint still consents to receive data.

Any packet received from the inverted 5-tuple refreshes consent if the packet is successfully validated by the protocol's authentication check (for instance, a MAC). Any DTLS message is sufficient to refresh consent, since these contain a MAC. For DTLS-SRTP [RFC5764], receipt of an authenticated SRTP packet is sufficient.

Consent is ended immediately by receipt of a an authenticated message that closes the connection (for instance, a TLS fatal alert).

Receipt of an unauthenticated end-of-session message (e.g., TCP FIN) does not indicate loss of consent. Thus, an endpoint receiving an unauthenticated end-of-session message SHOULD continue sending media (over connectionless transport) or attempt to re-establish the connection (over connection-oriented transport) until consent expires or it receives an authenticated message revoking consent.

2.1. Consent in WebRTC

WebRTC applications MUST cease transmission on a connection if they have not received any valid, authenticated packets for 30 seconds.

WebRTC applications that intend to maintain consent MUST send an authenticated packet at least every 10 seconds. If there is no application data to send, the DTLS heartbeat extension [RFC6520] MUST be sent to maintain consent. This reduces the probability that transient network failures cause consent expiration.

2.2. The Role of ICE

Given that DTLS is used to establish and maintain consent, ICE is only used to test and nominate candidate pairs. This allows for the use of DTLS without ICE, though this is unlikely to work for endpoints with poor connectivity.

If ICE is not employed, a DTLS server SHOULD use the denial of service countermeasures described in Section 4.2.1 of [RFC6347]; specifically the "HelloVerifyRequest" and the cookie that it carries.

2.3. Relationship with Connection Liveness

A connection is considered "live" if packets are received from a remote endpoint within an application-dependent period.

A WebRTC application can request a notification when there are no packets received for a certain period. Similarly, an application can request that heartbeats are sent after an interval shorter than 10 seconds. These two time intervals might be controlled by the same configuration item.

Sending heartbeats at a high rate could allow a malicious application to generate congestion. A WebRTC application MUST NOT be able to send heartbeats at a rate higher than 1 per second.

3. Application Layer Protocol Identifiers

The following ALPN identifiers are defined:

RTP (0x52 0x54 0x50): This token indicates that DTLS-SRTP [RFC5764] is acceptable or selected.

SCTP (0x53 0x43 0x54 0x50): This token indicates that WebRTC Data Channels [I-D.ietf-rtcweb-data-channel] is acceptable or accepted. The DTLS record-layer carries encapsulated Stream Control Transmission Protocol (SCTP) [RFC4960] packets as described in [I-D.ietf-tsvwg-sctp-dtls-encaps].

RTP+SCTP (0x52 0x54 0x50 0x2b 0x53 0x43 0x54 0x50): This token indicates that both DTLS-SRTP [RFC5764] and WebRTC Data Channels [I-D.ietf-rtcweb-data-channel] are acceptable or selected. The DTLS record-layer carries encapsulated SCTP packets as described in [I-D.ietf-tsvwg-sctp-dtls-encaps]; this is multiplexed with SRTP [RFC3711] packets as described in [RFC5764].

An application that can use a multiplexed combination of SRTP and SCTP MUST select "RTP+SCTP" if it is available, even if it is not using both protocols initially. This avoids any need to renegotiate application layer protocols as usage needs change.

4. Security Considerations

This document defines a security mechanism.

Consent does not establish any bounds on the volume of packets that a receiver is willing to accept. A receiver that receives packets at a rate in excess of what it is willing to tolerate can close the connection. If the close message is lost, this can result in

unwanted data being received until consent expires (i.e., 30 seconds).

SRTP is encrypted and authenticated with symmetric keys; that is, both sender and receiver know the keys. With two party sessions, receipt of an authenticated packet from the single remote party is a strong assurance the packet came from that party. However, when a session involves more than two parties, all of whom know each others keys, any of those parties could have sent (or spoofed) the packet. Such shared key distributions are possible with some MIKEY [RFC3830] modes, Security Descriptions [RFC4568], and EKT [I-D.ietf-avtcore-srtp-ekt].

5. IANA Considerations

This document registers three identifiers in the "Application Layer Protocol Negotiation (ALPN) Protocol IDs" established by [I-D.ietf-tls-applayerprotoneg].

Protocol: RTP over DTLS-SRTP

Identification Sequence: 0x52 0x54 0x50 ("RTP")

Specification: This document.

Protocol: WebRTC Data Channels

Identification Sequence: 0x53 0x43 0x54 0x50 ("SCTP")

Specification: This document.

Protocol: RTP over DTLS-SRTP multiplexed with WebRTC Data Channels

Identification Sequence: 0x52 0x54 0x50 0x2b 0x53 0x43 0x54 0x50 ("RTP+SCTP")

Specification: This document.

6. Acknowledgements

Muthu Arul Mozhi Perumal, Ram Mohan Ravindranath, Tirumaleswar Reddy, and Dan Wing are the authors of the original draft that dealt with managing consent.

7. References

7.1. Normative References

[I-D.ietf-rtcweb-data-channel]
Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Data Channels", draft-ietf-rtcweb-data-channel-06 (work in progress), October 2013.

[I-D.ietf-tls-applayerprotoneg]
Friedl, S., Popov, A., Langley, A., and S. Emile, "Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension", draft-ietf-tls-applayerprotoneg-03 (work in progress), October 2013.

[I-D.ietf-tsvwg-sctp-dtls-encaps]
Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "DTLS Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-dtls-encaps-02 (work in progress), October 2013.

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, February 2012.

7.2. Informative References

[I-D.ietf-avtcore-srtp-ekt]
McGrew, D. and D. Wing, "Encrypted Key Transport for Secure RTP", draft-ietf-avtcore-srtp-ekt-01 (work in progress), October 2013.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

Authors' Addresses

Martin Thomson
Mozilla
Suite 300
650 Castro Street
Mountain View, CA 94041
US

Email: martin.thomson@gmail.com

Dan Wing
Cisco Systems, Inc.
510 McCarthy Blvd.
Milpitas, CA 95035
US

Phone: (408) 853 4197
Email: dwing@cisco.com

Cullen Jennings
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1 408 421-9990
Email: fluffy@cisco.com