

RTCWEB
Internet-Draft
Intended status: Standards Track
Expires: November 20, 2014

M. Perumal
D. Wing
R. Ravindranath
T. Reddy
Cisco Systems
M. Thomson
Mozilla
May 19, 2014

STUN Usage for Consent Freshness
draft-ietf-rtcweb-stun-consent-freshness-03

Abstract

To prevent sending excessive traffic to an endpoint, periodic consent needs to be obtained from that remote endpoint.

This document describes a consent mechanism using a new STUN usage. This same mechanism can also determine connection loss ("liveness") with a remote peer.

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 November 20, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	Design Considerations	3
4.	Solution Overview	3
4.1.	Expiration of Consent	3
4.2.	Immediate Revocation of Consent	4
5.	Connection Liveness	5
6.	DiffServ Treatment for Consent packets	5
7.	W3C API Implications	5
8.	Security Considerations	6
9.	IANA Considerations	6
10.	Acknowledgement	6
11.	References	6
11.1.	Normative References	6
11.2.	Informative References	7
Appendix A.	Example Implementation	7
	Authors' Addresses	8

[1.](#) Introduction

To prevent attacks on peers, RTP endpoints have to ensure the remote peer wants to receive traffic. This is performed both when the session is first established to the remote peer using ICE connectivity checks, and periodically for the duration of the session using the procedures defined in this document.

When a session is first established, WebRTC implementations are required to perform STUN connectivity checks as part of ICE [[RFC5245](#)]. That initial consent is not described further in this document and it is assumed that ICE is being used for that initial consent.

Related to consent is loss of connectivity ("liveness"). Many applications want notification of connection loss to take appropriate actions (e.g., alert the user, try switching to a different interface).

This document describes a new STUN usage with a request and response messages which verifies the remote peer's consent to receive traffic, and can also detect loss of liveness.

2. Terminology

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 [\[RFC2119\]](#).

Consent: It is the mechanism of obtaining permission to send traffic to a certain transport address. This is the initial consent to send traffic, which is obtained by ICE or a TCP handshake.

Consent Freshness: Permission to continue sending traffic to a certain transport address. This is performed by the procedure described in this document.

Session Liveness: Detecting loss of connectivity to a certain transport address. This is performed by the procedure described in this document.

Transport Address: The remote peer's IP address and (UDP or TCP) port number.

3. Design Considerations

Although ICE requires periodic keepalive traffic to keep NAT bindings alive ([Section 10 of \[RFC5245\]](#), [\[RFC6263\]](#)), those keepalives are sent as STUN Indications which are send-and-forget, and do not evoke a response. A response is necessary both for consent to continue sending traffic, as well as to verify session liveness. Thus, we need a request/response mechanism for consent freshness. ICE can be used for that mechanism because ICE already requires ICE agents continue listening for ICE messages, as described in [section 10 of \[RFC5245\]](#).

4. Solution Overview

There are two ways consent to send traffic is revoked: expiration of consent and immediate revocation of consent, which are discussed in the following sections.

4.1. Expiration of Consent

A WebRTC browser performs a combined consent freshness and session liveness test using STUN request/response as described below:

An endpoint **MUST NOT** send application data (e.g., RTP, RTCP, SCTP, DTLS) on an ICE-initiated connection unless the receiving endpoint consents to receive the data. After a successful ICE connectivity check on a particular transport address, subsequent consent **MUST** be

obtained following the procedure described in this document. The consent expires after a fixed amount of time.

Explicit consent to send is indicated by sending an ICE binding request to the remote peer's Transport Address and receiving a matching and authenticated ICE binding response from the inverted remote peer's Transport Address. These ICE binding requests and responses are authenticated using the same short-term credentials as the initial ICE exchange, but using a new (fresh) transaction-id each time consent needs to be refreshed. Implementations **MUST** obtain fresh consent before their existing consent expires. If an ICE binding response is not received within 1.5 times the previous round trip time, another ICE binding request is sent using the a new (fresh) transaction-id (so that round-trip time can be calculated), and re-transmissions **MUST NOT** be sent more frequently than every 500ms or the smoothed round-trip time (from previous consent freshness checks or RTP round-trip time), whichever is less. For the purposes of this document, receipt of an ICE response with the matching transaction-id of its request with a valid MESSAGE-INTEGRITY is considered a consent response.

The initial Consent to send traffic is obtained by ICE. Consent expires after 30 seconds. That is, if a valid STUN binding response corresponding to one of the STUN requests sent in the last 30 seconds has not been received from the inverted 5-tuple, the endpoint **MUST** cease transmission on that 5-tuple.

To meet the security needs of consent, an untrusted application (e.g., JavaScript) **MUST NOT** be able to obtain or control the ICE transaction-id, because that enables spoofing STUN responses, falsifying consent

An endpoint that is only receiving application traffic (recvonly) does not need to obtain consent which can slightly conserve its resources. However, the endpoint needs to ensure its NAT or firewall mappings persist which can be done using keepalive or other techniques (see [Section 10 of \[RFC5245\]](#) and see [\[RFC6263\]](#)). If the endpoint wants send application traffic, it needs to first obtain consent if its consent expired.

4.2. Immediate Revocation of Consent

The previous section explained how consent expires due to a timeout. In some cases it is useful to signal a connection is terminated, rather than relying on a timeout. This is done by immediately revoking consent.

Consent for sending traffic on the media or data channel is revoked by receipt of an authenticated message that closes the connection (for instance, a TLS fatal alert).

Receipt of an unauthenticated message that closes a connection (e.g., TCP FIN) does not indicate revocation 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.

5. Connection Liveness

A connection is considered "live" if packets are received from a remote endpoint within an application-dependent period. An application can request a notification when there are no packets received for a certain period (configurable).

Similarly, if packets haven't been received within a certain period, an application can request a consent check (heartbeat) be generated. These two time intervals might be controlled by the same configuration item.

Sending consent checks (heartbeats) at a high rate could allow a malicious application to generate congestion, so applications MUST NOT be able to send heartbeats faster than 1 per second.

6. DiffServ Treatment for Consent packets

It is RECOMMENDED that STUN consent checks use the same Diffserv Codepoint markings as the media packets sent on that transport address. This follows the recommendation of ICE connectivity check described in [section 7.1.2.4 of \[RFC5245\]](#).

Note: It is possible that different Diffserv Codepoints are used by different media over the same transport address [[I-D.ietf-tsvwg-rtcweb-qos](#)]. In that case, what should this document recommend as the Codepoint for STUN consent packets ?

7. W3C API Implications

For the consent freshness and liveness test the W3C specification should provide APIs as described below:

1. Ability for the browser to notify the JavaScript that a consent freshness transaction has failed for a media stream and the browser has stopped transmitting for that stream.

2. Ability for the JavaScript to start and stop liveness test and set the liveness test interval.
3. Ability for the browser to notify the JavaScript that a liveness test has failed for a media stream.

8. Security Considerations

This document describes a security mechanism.

The security considerations discussed in [[RFC5245](#)] should also be taken into account.

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](#)]. Thus, in such shared keying distributions, receipt of an authenticated SRTP packet is not sufficient.

9. IANA Considerations

This document does not require any action from IANA.

10. Acknowledgement

Thanks to Eric Rescorla, Harald Alvestrand, Bernard Aboba, Magnus Westerland, Cullen Jennings and Simon Perreault for their valuable inputs and comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.

- [RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", [RFC 6263](#), June 2011.

11.2. Informative References

- [I-D.ietf-avtcore-srtp-ekt]
McGrew, D. and D. Wing, "Encrypted Key Transport for Secure RTP", [draft-ietf-avtcore-srtp-ekt-02](#) (work in progress), February 2014.
- [I-D.ietf-tsvwg-rtcweb-qos]
Dhesikan, S., Druta, D., Jones, P., and J. Polk, "DSCP and other packet markings for RTCWeb QoS", [draft-ietf-tsvwg-rtcweb-qos-00](#) (work in progress), April 2014.
- [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.

Appendix A. Example Implementation

This section describes one possible implementation algorithm of consent. This section is non-normative and provided for reference only.

The solution uses three values:

1. A consent timer, T_c , whose value is set to 30 seconds.
2. A packet receipt timer, T_r , whose value is determined by the application. T_r can be greater than 1 but less than 30 seconds and has a default value of 5 seconds.
3. A consent timeout, T_f , which is how many seconds elapse without a consent response before the browser ceases transmission of media. Its value is be 30 seconds or less.
4. A retransmission Timer, T_{ret} , whose value is determined by the RTT of a given path. The duration of this timer is set to 1.5 times of (500 ms or the smoothened round-trip time (from previous consent freshness checks or RTP round-trip time)), whichever is less.

A WebRTC browser performs a combined consent freshness and session liveness test using STUN request/response as described below:

Every T_c seconds, the WebRTC browser sends a STUN Binding Request to the peer. The difference from ICE connectivity check is that there is no exponential back off for retransmissions.

If a valid STUN Binding Response is received, the consent timer is reset to the time of receiving the response and fires again T_c seconds later.

If a valid STUN Binding Response is not received after T_{ret} milliseconds, the STUN Binding Request is retransmitted (with a new Transaction ID). As long as a valid STUN Binding Response is not received, this retransmission is repeated every T_{ret} milliseconds until T_f seconds have elapsed or a valid response is received. If no valid response is received after T_f seconds, the WebRTC browser quits transmitting traffic to this remote peer. The streams that are being sent on a flow(5-tuple) for which a consent has failed will be stopped. If the default value of T_f is 30 seconds then media transmission will stop Consent (T_f) expires.

Authors' Addresses

Muthu Arul Mozhi Perumal
Cisco Systems
Cessna Business Park
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: mperumal@cisco.com

Dan Wing
Cisco Systems
821 Alder Drive
Milpitas, California 95035
USA

Email: dwing@cisco.com

Ram Mohan Ravindranath
Cisco Systems
Cessna Business Park
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: rmohanr@cisco.com

Tirumaleswar Reddy
Cisco Systems
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

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

Email: martin.thomson@gmail.com

