Behave
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

Muthu. Perumal
D. Wing
R. Ravindranath
T. Reddy
Cisco Systems
July 15, 2013

STUN Usage for Consent Freshness
draft-muthu-behave-consent-freshness-04

Abstract

Verification of peer consent before sending traffic is necessary in WebRTC deployments to ensure that a malicious JavaScript cannot use the browser as a platform for launching attacks. A related problem is session liveness. WebRTC application may want to detect connection failure and take appropriate action.

This document describes how a WebRTC browser can verify peer consent to continue sending traffic and detect connection failure.

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 16, 2014.

Copyright Notice

Table of Contents

1. Introduction

To prevent attacks on WebRTC peers, WebRTC browsers 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 when for the duration of the session (using the procedure 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.

Related to consent is loss of connectivity ("liveness"). WebRTC applications want notification of connection failure 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 which verifies the remote peer consents to receive traffic, and detects loss of liveness. To meet the security needs of consent, the JavaScript application has no control over the consent requests or the requirement to receive a consent response. However, the

JavaScript does get notification of consent failure and loss of connectivity.

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 from the peer to send traffic on a candidate pair.

Consent Freshness: It is the mechanism of obtaining permission from the peer to continue sending traffic on a nominated candidate pair after ICE has concluded.

Session Liveness: It is the mechanism of detecting connectivity on a nominated candidate pair after ICE has concluded.

Transport Address: The combination of an IP address and port number (such as a UDP or TCP port number).

3. Design Considerations

Although ICE requires periodic keepalive traffic to be sent in order to keep NAT bindings alive (Section 10 of [RFC5245], [RFC6263]), those keepalives 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 ensure connectivity. Thus, we need a request/response mechanism.

Though ICE specifies STUN Binding indications to be used for keepalives, it requires that an agent be prepared to receive connectivity check as well. If a connectivity check is received, a response is generated, but there is no impact on ICE processing, as described in section 10 of [RFC5245].

While a WebRTC browser could verify whether the peer continues to send SRTCP reports before sending traffic to the peer, the usage of SRTCP together with Security Descriptions [RFC4568] requires exposing the media keys to the JavaScript and renders SRTCP unsuitable for

consent freshness.

For consent, calculating the SHA1 HMAC is necessary for MESSAGE-INTEGRITY which is computationally expensive. Security analysis concluded that the STUN 96 bits transaction ID is sufficient for consent, without needing MESSAGE-INTEGRITY. However, omitting the MESSAGE-INTEGRITY attribute from STUN Binding request/response to avoid the cost of computing SHA1 would make browsers incapable of verifying consent freshness with legacy ICE/ICE-lite implementations, resulting in backward compatibility issues.

The above considerations suggest that STUN Binding request/response is most suitable for performing consent freshness.

4. Solution Overview

Consent freshness serves as a circuit breaker (so that if the path or remote peer fails the WebRTC browser stops sending all traffic on that remote peer), determining session liveness serves the purpose of notifying the application of connectivity failure so that the application can take appropriate action.

The solution uses three values:

1. A consent timer, Tc, whose value is determined by the browser. This value MUST be 15 seconds.

2. A packet receipt timer, Tr, whose value is determined by the application, but MUST NOT be shorter than 1 second or longer than 15 seconds, and SHOULD have a default value of 5 seconds.

3. A consent timeout, Tf, which is how many seconds elapse without a consent response before the browser ceases transmission of media. Its value MUST be 15 seconds or less, and the value 15 seconds is RECOMMENDED.

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

Every Tc seconds, the WebRTC browser sends a STUN Binding Request to

the peer. This request MUST use a new, cryptographically random Transaction ID [RFC4086], and is formatted as for an ICE connectivity check [RFC5245]. A valid STUN Binding Response is also formatted as for an ICE connectivity check [RFC5245]. The STUN Binding Request and STUN Binding Response are validated as for an ICE connectivity check [RFC5245].

If a valid STUN Binding Response is received, the consent timer is reset and fires again Tc seconds later.

If a valid STUN Binding Response is not received after 500ms, the STUN Binding Request is retransmitted (with the same Transaction ID and all other fields). As long as a valid STUN Binding Response is not received, this retransmission is repeated every 500ms until Tf seconds have elapsed or a valid response is received. If no valid response is received after Tf seconds, the WebRTC browser MUST quit transmitting traffic to this remote peer. Considering the default value of Tf=15 seconds, this means transmission will stop after 30 consent check packets have resulted in no response.

Liveness timer: If no packets have been received on the local port in Tr seconds, the WebRTC browser MUST inform the JavaScript that connectivity has been lost. The JavaScript application will use this notification however it desires (e.g., cease transmitting to the remote peer, provide a notification to the user, etc.). Note the definition of a received packet is liberal, and includes an SRTP packet that fails authentication, a STUN Binding Request with an invalid USERNAME or PASSWORD, or any other packet.

5. 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.

6. Interaction with Keepalives used for Refreshing NAT Bindings

When not actively sending traffic on a nominated candidate pair, performing consent freshness does not serve any purpose from a security perspective. If consent freshness is not performed during this period, the browser continues to performs the ICE keepalives [RFC5245] or RTP keepalive [RFC6263] to refresh NAT bindings.

7. Security Considerations

Security considerations discussed in [RFC5245] are to be taken into account.

In ICE [RFC5245] the STUN request/response are protected with MESSAGE-INTEGRITY, using an ephemeral username and password exchanged in the SDP ice-ufrag and ice-pwd attributes. This prevents ICE from accidentally connecting to an in-intended peer, in that ICE will only connect to a peer that also knows the same username and password (exchanged in call signaling). Once that connection to the remote peer has been established with ICE, the consent to continue sending traffic does not benefit from re-asserting that same username and password, so long as the senders and receiver's IP addresses remain the same (as they usually do).

8. IANA Considerations

This document does not require any action from IANA.

9. Acknowledgement

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

10. Normative References

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

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

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

[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.

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