Behave
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

Muthu A M. Perumal
D. Wing
R. Ram Mohan
Cisco Systems
H. Kaplan
Acme Packet

July 16, 2012

STUN Usage for Consent Freshness and Session Liveness
draft-muthu-behave-consent-freshness-01

Abstract

Verification of peer consent 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 applications may want to detect connection failure and take appropriate actions. This document describes a STUN usage that enables a WebRTC browser to perform the following on a candidate pair ICE is using for a media component after session establishment:

1. Verify the peer consent for continuing to receive traffic.
2. Dectect connection failure.

This also serves the purpose of refreshing NAT bindings.

Status of this Memo

This Internet-Draft is submitted to IETF 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 17, 2013.

Table of Contents

1. Introduction

Consent verification is the mechanism using which WebRTC implementations can verify the peer consent for receiving traffic on candidate media transport addresses. This has two parts

1. Verifying peer consent for receiving traffic on candidate media transport addresses at session establishment.
2. Verifying peer consent for continuing to receive traffic on candidate media transport addresses after session establishment.

WebRTC implements are required to perform STUN connectivity checks at session establishment as part of ICE procedures [RFC5245]. This takes care of the first part of the consent verification described above.

After session establishment ICE requires STUN Binding indications to be used for refreshing NAT bindings for a candidate pair ICE is using for a media component. Since a STUN Binding indication does not evoke a response, it cannot be used for the second part of the consent verification describes above.

A related problem is session liveness. WebRTC applications may want to detect connection failure on candidate media transport addresses after session establishment and take appropriate actions. Again, the STUN Binding indications in ICE sent after session establishment cannot be used for determining session liveness.

This document describes a STUN usage based on STUN request/response that enables a WebRTC browser to perform the following on a candidate pair ICE is using for a media component after session establishment:

1. Verify the peer consent for continuing to receive traffic.
2. Dectect connection failure.

This also serves the purpose of refreshing NAT bindings.

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].

3. Definitions

Consent Freshness: It is the mechanism of verifying peer consent for continuing to receive traffic on a candidate pair ICE is using for a media component after ICE has concluded. This document uses completion of session establishment synonymous with the conclusion of ICE.

Session Liveness: It is the mechanism of detecting connectivity on a candidate pair ICE is using for a media component after ICE has concluded.

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

4. Solution Overview

The solution uses two timers:

1. A consent timer Tc whose value is determined by the browser.
2. A packet receipt timer whose value is determined by the application.

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

- Starts a consent timer Tc (no less than 15 sec).
- Starts a packet receipt timer Tr (no less than 500 msec); application configurable.
- When either timer expires it starts a STUN transaction.
- When the STUN transaction succeeds, it re-starts both timers.
- When the STUN transaction fails
  - If the transaction was started by timer Tc, it stops sending traffic on that candidate pair.
  - Else, it notifies the application of the failure and continues.
- It resets timer Tr on receiving any packet from the other side.

While consent freshness serves as a circuit breaker (if there is a failure the WebRTC browser stops sending all traffic on that candidate pair), determining session liveness serves the purpose of notifying the application of connectivity failure so that the application can take appropriate action.

5. Design Considerations

As described earlier in this document, STUN indications are not suitable for performing consent freshness. Hence, performing consent freshness requires the use of STUN request/response.

ICE requires the usage of message integrity with STUN using its short-term credential mechanism. The need for this mechanism goes beyond just security and is required for the correct operation of the ICE connectivity check procedures; without message integrity the connectivity checks can yield false positives, as described in Appendix B section B.4 of RFC5245. However, this problem is not applicable for consent freshness, since consent freshness is performed only after ICE concludes.

One of the reasons for ICE choosing STUN Binding indications for keepalives is because Binding indication allows integrity to be disabled, allowing for better performance. This is useful for large-scale endpoints, such as PSTN gateways and SBCs as described in Appendix B section B.10 of RFC5245

STUN requires the 96 bits transaction ID to be uniformly and randomly chosen from the interval 0 .. 2**96-1, and be cryptographically random. This is good enough security against an off-path attacker.

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.

Reusing STUN Binding request/response allows browsers to interoperate with existing ICE implementations; even ICE-lite implementations. This is considered very important.

Conclusion: Considering all the above, there seems to be a rough consensus in the RTCWEB WG for reusing the STUN Binding request/ response for determining consent freshness and session liveness. The current thought is that the cost of the SHA-1 computation is not a good enough justification for the pain a new method would cause with existing IEC implementations.

6. STUN Consent Method

The STUN message type field from the STUN specification [RFC5389] is shown below

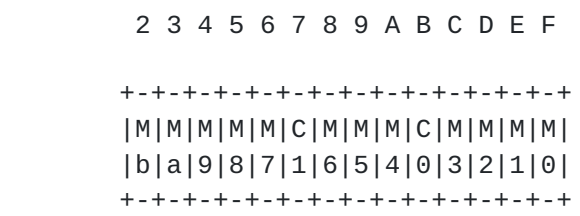

Figure 1: Format of STUN Message Type Field

Here the bits in the message type field are shown as most significant (Mb) through least significant (M0). C1 and C0 represent a 2-bit encoding of the class. Mb through M0 represent a 12-bit encoding of the method. The Consent method is encoded as 0b000000000010.

A Consent request has class=0b00 (request) and method=0b000000000010 (Consent) and is encoded into the first 16 bits of the STUN header as 0x0002.

A Consent success response has class=0b10 (success response) and method=0b000000000010 (Consent) and is encoded into the first 16 bits of the STUN header as 0x0102.

A Consent error response has class=0b11 (error response) and method=0b000000000010 (Consent) and is encoded into the first 16 bits of the STUN header as 0x0112.

A Consent indication has class=0b01 (indication) and method=0b000000000010 (Consent) and is encoded into the first 16 bits of the STUN header as 0x0012.

7. STUN Consent Method Processing

Processing of the STUN Consent method is similar to the processing of the STUN Binding method except that the procedures pertaining to the message integrity and short/long-term credential mechanisms are not applicable. In particular, the USERNAME and MESSAGE-INTEGRITY attributes are not included in a Consent request or response.

7.1. Generating a Consent Request

TBD

7.2. Receiving a Consent Request

TBD

7.3. Generating a Consent Response

TBD

7.4. Receiving a Consent Response

TBD

8. Performing Consent Freshness

TBD

9. Examples

TBD

10. Generating Consent Freshness Response

A STUN agent receiving a consent freshness request for a candidate pair ICE is using for a media component MUST generate a STUN Consent response.

11. SDP Extension for Consent Freshness

TBD

12. Interaction with Keepalives used for Refreshing NAT Bindings

An implementation that performs the procedures described in this document has no need to also perform the keepalives described in ICE [RFC5245] or RTP keepalive [RFC6263], as they both force recurring messages to be sent over the UDP port used by RTP. Thus, an implementation that performs the procedures described in this document SHOULD NOT also do the keepalives described in ICE [RFC5245] or RTP keepalives [RFC6263] for the UDP port used for RTP.

13. Open Items

1. Should STUN Binding request/response be reused for determining consent freshness and session liveness, considering interoperability with existing ICE and ICE-lite implementation even at the cost of the incurred SHA-1 computation?

2. If the RTCP port is different from the RTP port, does RTCP need consent freshness and session liveness tests?

14. Security Considerations

TBD

15. IANA Considerations

TBD

16. Acknowledgement

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

17. Normative References

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

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[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.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

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

Email: rmohanr@cisco.com

Hadriel Kaplan
Acme Packet

Email: hkaplan@acmepacket.com