RTCWEB M. Perumal
Internet-Draft Ericsson
Intended status: Standards Track D. Wing
Expires: November 14, 2015 R. Ravindranath
T. Reddy
Cisco Systems
M. Thomson
Mozilla
May 13, 2015

STUN Usage for Consent Freshness
draft-ietf-rtcweb-stun-consent-freshness-13

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 Session Traversal Utilities for NAT (STUN) usage.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

To prevent attacks on peers, endpoints have to ensure the remote peer is willing to receive traffic. This is performed both when the session is first established to the remote peer using Interactive Connectivity Establishment ICE [RFC5245] connectivity checks, and periodically for the duration of the session using the procedures defined in this document.

When a session is first established, ICE implementations obtain an initial consent to send by performing STUN connectivity checks. This document describes a new STUN usage with exchange of request and response messages that verifies the remote peer's ongoing consent to receive traffic. This consent expires after a period of time and needs to be continually renewed, which ensures that consent can be terminated.

This document defines what it takes to obtain, maintain, and lose consent to send. Consent to send applies to a single 5-tuple. How applications react to changes in consent is not described in this document.

Consent is obtained only by full ICE implementations. An ICE-lite implementation will not generate consent checks, but will just respond to consent checks it receives. No changes are required to ICE-lite implementations in order to respond to consent checks, as they are processed as normal ICE connectivity checks.

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: The mechanism of obtaining permission to send to a remote transport address. Initial consent is obtained using ICE.

Consent Freshness: Maintaining and renewing consent over time.

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 for consent to continue sending traffic. Thus, we need a request/response mechanism for consent freshness. ICE can be used for that mechanism because ICE implementations are already required to continue listening for ICE messages, as described in section 10 of [RFC5245]. If consent is performed then there is no need to send keepalive messages.

4. Solution

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 full ICE implementation performs consent freshness test using STUN request/response as described below:

An endpoint MUST NOT send data other than paced STUN connectivity checks or responses toward any transport address unless the receiving endpoint consents to receive data. That is, no application data (e.g., RTP or DTLS) can be sent until consent is obtained. After a successful ICE connectivity check on a particular transport address,

consent MUST be maintained following the procedure described in this document.

Explicit consent to send is obtained and maintained by sending an STUN binding request to the remote peer's transport address and receiving a matching, authenticated, non-error STUN binding response from the remote peer's transport address. These STUN binding requests and responses are authenticated using the same short-term credentials as the initial ICE exchange.

Note: Although TCP has its own consent mechanism (TCP acknowledgements), consent is necessary over a TCP connection because it could be translated to a UDP connection (e.g., [RFC6062]).

Initial consent to send traffic is obtained using ICE. Consent expires after 30 seconds. That is, if a valid STUN binding response corresponding to any STUN request sent in the last 30 seconds has not been received from the remote peer's transport address, the endpoint MUST cease transmission on that 5-tuple. STUN consent responses received after consent expiry do not re-establish consent, and may be discarded or cause an ICMP error.

To prevent expiry of consent, a STUN binding request can be sent periodically. To prevent synchronization of consent checks, each interval MUST be randomized from between 0.8 and 1.2 times the basic period. Implementations SHOULD set a default interval of 5 seconds, resulting in a period between checks of 4 to 6 seconds.

Each STUN binding request for consent MUST use a new cryptographically strong [RFC4086] STUN transaction ID. Each STUN binding requests for consent is transmitted once only. Hence, the sender cannot assume that it will receive a response for each consent request, and a response might be for a previous request (rather than for the most recently sent request). Consent expiration causes immediate termination of all outstanding STUN consent transactions. Each STUN transaction is maintained until one of the following criteria is fulfilled:

- A STUN response associated with the transaction is received; or

- A STUN response associated to a newer transaction is received.

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

To prevent attacks on the peer during ICE restart, an endpoint that continues to send traffic on the previously validated candidate pair during ICE restart MUST continue to perform consent freshness on that candidate pair as described earlier.

While TCP affords some protection from off-path attackers ([RFC5961], [RFC4953]), there is still a risk an attacker could cause a TCP sender to send forever by spoofing ACKs. To prevent such an attack, consent checks MUST be performed over all transport connections, including TCP. In this way, an off-path attacker spoofing TCP segments can not cause a TCP sender to send once the consent timer expires (30 seconds).

An endpoint that is not sending any application data does not need to maintain consent. However, not sending any traffic could cause NAT or firewall mappings to expire. Furthermore, having one peer unable to send is detrimental to many protocols. Absent better information about the network, if an endpoint needs to ensure its NAT or firewall mappings do not expire, it can be done using keepalive or other techniques (see Section 10 of [RFC5245] and see [RFC6263]).

After consent is lost for any reason, the same ICE credentials MUST NOT be used on the affected 5-tuple again. That means that a new session, or an ICE restart, is needed to obtain consent to send.

4.2. Immediate Revocation of Consent

In some cases it is useful to signal that consent is terminated rather than relying on a timeout.

Consent for sending application data is immediately revoked by receipt of an authenticated message that closes the connection (e.g., a TLS fatal alert) or receipt of a valid and authenticated STUN response with error code Forbidden (403). Note however that consent revocation messages can be lost on the network, so an endpoint could resend these messages, or wait for consent to expire.

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.

Note that an authenticated SRTCP BYE does not terminate consent; it only indicates the associated SRTP source has quit.

5. DiffServ Treatment for Consent

It is RECOMMENDED that STUN consent checks use the same Diffserv Codepoint markings as the ICE connectivity checks described in Section 7.1.2.4 of [RFC5245] for a given 5-tuple.

Note: It is possible that different Diffserv Codepoints are used by different media over the same transport address [I-D.ietf-tsvwg-rtcweb-qos]. Such a case is outside the scope of this document.

6. DTLS applicability

The DTLS applicability is identical to what is described in Section 4.2 of [RFC7350].

7. API Recommendations

The W3C specification [W3C-WEBRTC] may provide an API hook that generates an event when consent has expired for a given 5-tuple, meaning that transmission of data has ceased. This could indicate what application data is affected, such as media or data channels.

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 to verify consent.

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, Christer Holmberg, Simon Perreault, Paul Kyzivat, Emil Ivov, Jonathan Lennox, Inaki Baz Castillo, Rajmohan Banavi and Christian Groves for their valuable inputs and comments. Thanks to Christer Holmberg for doing a through review.

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.

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

[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] Mattsson, J., McGrew, D., and D. Wing, "Encrypted Key Transport for Secure RTP", draft-ietf-avtcore-srtp-ekt-03 (work in progress), October 2014.

[I-D.ietf-rtcweb-overview] Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-13 (work in progress), November 2014.

[I-D.ietf-tsvwg-rtcweb-qos] Dhesikan, S., Jennings, C., Druta, D., Jones, P., and J. Polk, "DSCP and other packet markings for RTCWeb QoS", draft-ietf-tsvwg-rtcweb-qos-03 (work in progress), November 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.

[RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks", RFC 4953, July 2007.

[RFC5961] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks", RFC 5961, August 2010.

[RFC6062] Perreault, S. and J. Rosenberg, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations", RFC 6062, November 2010.

[RFC7350] Petit-Huguenin, M. and G. Salgueiro, "Datagram Transport Layer Security (DTLS) as Transport for Session Traversal Utilities for NAT (STUN)", RFC 7350, August 2014.

[W3C-WEBRTC]
Bergkvist, A., Burnett, D., Narayanan, A., and C. Jennings, "WebRTC 1.0: Real-time Communication Between Browsers", february 2015.

Authors' Addresses

Muthu Arul Mozhi Perumal
Ericsson
Ferns Icon
Doddanekundi, Mahadevapura
Bangalore, Karnataka 560037
India

Email: muthu.arul@gmail.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