Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2016
October 14, 2015

B. Williams
Akamai
J. Uberti
Google

Ufrag Permissions for Traversal Using Relays around NAT (TURN)
draft-williams-tram-ufrag-permission-00

Abstract

When using a TURN relay, ICE connectivity checks require an explicit permission or channel binding to be established for each peer address to be checked. This requires the answerer to send its candidate addresses to the offerer via the rendezvous server, which can impose a latency penalty when the rendezvous server is centrally located. This document defines a new type of TURN permission that will allow any ICE connectivity check message that contains the offerer's ufrag value to be accepted on a relay address for delivery over the associated TURN tunnel.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Interactive Connectivity Establishment (ICE) [RFC5245] provides a connectivity checking mechanism that peers can use to determine how to communicate directly with each other (e.g. which network layer protocol to use, which network address and transport port to use, etc.). The peers gather their sets of candidate addresses and exchange them via a rendezvous server using an offer/answer protocol. After gathering the addresses, the peers then send connectivity checks between address pairs to find a suitable local/remote adress pair to use for communication.

Successful direct connectivity checks between the peers is the simplest scenario.

```
                              +------------+
                              |            |
         +--------------------> rendezvous +---------------------+
         |                    |   server   | 2                   |
         |                    |            |                     |
         |                    +------------+                     |
         |                                                       |
         |                                                       |
         |                                                       |
         |                                                       |
         | 1                                                     |
         |                                                       |
   +-----+------+                                         +------v-----+
   |            |                                       3 |            |
   |  offerer   <-----------------------------------------+  answerer  |
   |            |                                         |            |
   +------------+                                         +------------+
```

The offerer sends an offer with its candidate addresses to the rendezvous server (1), the rendezvous server forwards the offer to the answerer (2), and the answerer is able to send a connectivity check directly to the offerer (3) at the same time that it sends its answer back to the offerer via the rendezvous server.

Successful connectivity checks for a relayed candidate with Traversal Using Relays around NAT (TURN) [RFC5766] is more complicated and time consuming, partially due to the requirement for the local peer to explicitly notify the relay server about every permitted remote address.

```
                              +------------+
                              |            | 2
         +--------------------> rendezvous +---------------------+
         |--------------------+   server   <---------------------|
         ||                 4 |            |                    ||
         ||                   +------------+                    ||
         ||                                                     ||
         ||                                                     ||
         ||                                                     ||
         ||                                                     ||
         ||                                                     ||
       1 ||                                                   3 ||
   +------v-----+             +------------+              +------v-----+
   |            | 5           |            |              |            |
   |  offerer   +------------->   relay    |              |  answerer  |
   |            +------------->            +-------------->            |
   +------------+ 6           +------------+ 7            +------------+
```

In this case, the offerer first issues an allocation request to the relay server (not pictured) before sending an offer that includes the assigned relay address to the rendezvous server (1), which forwards the offer to the answerer (2). If the answerer sends a connectivity check to the relay address immediately, the relay will reject the message because there is no permission established for the answerer's address yet. Instead, the answerer must send its answer along with its candidate list to the rendezvous server (3), which relays the answer to the offerer (4). Only now can the offerer send a permission request to the relay (5) and then send a connectivity check message to the relay (6) to be forwarded to the answerer (7). Since the answer must be delivered before the necessary TURN permissions can be established, successful connectivity checks via the offerer's relay require an extra half round trip time via the rendezvous server as compared to direct host-to-host connectivity checks. This could be a significant penalty in the common case of a remotely located rendezvous server.

The latency penalty for the relay use case could be mitigated by permitting all ICE connectivity check messages to be delivered by the relay, regardless of whether there is an active permission for the sender. However, doing so would mean that use of the relay opens up the client to potential attacks from anywhere on the Internet. TURN permissions limit the risk by requiring the attacker to first discover an address associated with an active permission. This may be trivial to accomplish for an attacker who is on-path between the answerer and the relay, but would be more difficult for an off-path attacker.

When ICE is in use, the offer and answer messages each contain an ice-ufrag value, which is used in connectivity check messages as part of the username for Session Traversal Utilities for NAT (STUN) [RFC5389]. This document describes a new TURN permission type that allows any ICE connectivity check message to be relayed to TURN client if it has the expected remote ufrag (RFRAG) value in the STUN username attribute. This mechanism allows ICE connectivity checks to the offerer's relayed candidate to succeed without having to wait for the answer to arrive at the offerer, while at the same time continuing to require an attacker to learn some information about an active permission in order to construct packets that will be accepted by the relay dor delivery to the client.

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. Ufrag Permission Usage

To allow successful connetivity checks from the answerer, the offerer registers a new type of permission, known as a ufrag permission, with the relay server. Instead of using an XOR-PEER-ADDRESS attribute to identify the the remote peer, a ufrag permission specifies the offerer's ufrag as the value for a LOCAL-UFRAG attribute. A ufrag permission allows any ICE connectivity check from a remote peer to be accepted by the relay if the RFRAG in that message matches the ufrag specified for the permission. Note that the LOCAL-UFRAG attribute is only allowed for TURN permission requests. ChannelBind requests cannot make use of this type of permission.

Message flow for successful connectivity checks using ufrag permissions looks fairly similar to the direct connectivity case where timing of the first successful check is concerned.

```
                              +------------+
                              |            | 2
         +--------------------> rendezvous +---------------------+
         |                    |   server   |                     |
         |                    |            |                     |
         |                    +------------+                     |
         |                                                       |
         |                                                       |
         |                                                       |
         |                                                       |
         |                                                       |
       1 |                                                       |
   +-----+------+ 1'          +------------+              +------v-----+
   |            +-------------->            |              |            |
   |  offerer   |             |   relay    |              |  answerer  |
   |            <-------------+            <--------------+            |
   +------------+           4 +------------+            3 +------------+
```

The offerer first establishes a TURN allocation with the relay (not pictured) to learn its relay candidate(s). At the point when the offerer sends the offer to the rendezvous server (1), it also sends a ufrag permission request to the relay (1'). The rendezvous server forwards the offer to the answerer (2), at which point the answerer can immediately send ICE connectivity checks (3) that can be accepted by the relay and forwarded to the offerer (4). Provided that the relay is fairly close to the offerer or at least inline between the offerer and the answerer, the primary difference in latency between direct and relay connectivity checks is the time required for candidate gathering (i.e. the allocation request).

3.1. Forming a CreatePermission Request

A ufrag permission request is formed in the same general way as a permission associated with an IP address, with the only exception being that it contains a LOCAL-UFRAG attribute to provide the ufrag value. As with any other CreatePermission request, multiple permissions may be established using a single CreatePermission request, meaning that a combination of one or more XOR-PEER-ADDRESS attributes and one or more LOCAL-UFRAG attributes may be present in a single request, with each resulting in a separate permission.

The LOCAL-UFRAG attribute is an understanding required attribute with the type TBD, and it contains a single value, which is the sender's ufrag value. This is allowed to be from 4 to 256 bytes in length.

NOTE: The authors considered two alternatives: providing the ufrag in either an XOR-PEER-ADDRESS or a USERNAME attribute. In both cases, it this change would modify the semantics for the attribute enough that it seemed better to defined a new attribute type.

3.2. Processing a CreatePermission Request

When the server receives a CreatePermission request, it processes it as per [RFC5766]. The rest of this section describes processing for cases where the request contains a LOCAL-UFRAG attribute.

If the server understands but does not support ufrag addresses, it rejects the request with a 403 (Forbidden) error.

If the request is valid, then the server installs or refreshes a permission for the ufrag contained in the LOCAL-UFRAG attribute. The server then responds with a CreatePermission success response.

NOTE: Careful consideration of the ufrag permission's lifetime is required. It needs to be long enough to be useful for its intended purpose but short enough to limit security exposure. A future revision of the draft will cover this in more detail.

3.3. Server Backward Compatibility

A server that does not recognize the LOCAL-UFRAG attribute will reject the request and send a 420 (Unknown Attribute) error response and otherwise ignore the request.

If the request sent by the client contained IP address XOR-PEER-ADDRESS attributes in addition to a LOCAL-UFRAG attribute, the client MAY resend the request without the LOCAL-UFRAG attribute in order to retry registration of the IP address permissions.

3.4. Processing a ChannelBind Request

A ChannelBind request received on the server MUST be considered invalid if it contains a LOCAL-UFRAG attribute. The server MUST reject such a request with a 403 (Forbidden) error.

3.5. Processing a Message on the Relay Transport Address

When a message is received on the relay transport address, the server first checks whether the allocation has a matching IP/IPv6 permission. If it does not have a matching IP/IPv6 permission but it does have one or more ufrag permissions, the server examines the message to determine whether it is an ICE connectivity check message, meaning: it is a STUN Binding request that contains all of the required atrributes: FINGERPRINT, PRIORTY, ICE-CONTROLLED or ICE-CONTROLLING, USERNAME, and MESSAGE-INTEGRITY. If the message is not a structurally valid ICE connectivity check, the server MUST discard the message if there is no IP/IPv6 permission that applies.

If the message is an ICE connectivity check with no matching IP/IPv6 permission, the server then parses the value of the USERNAME attribute to extract the RFRAG value, which is the second colon-separated field. If a ufrag permission exists for the RFRAG value, the relay server generates a Data indication for the message. The Data indication is then sent to the TURN client.

NOTE: TURN-TCP [RFC6062] should be discussed in this document if/when it moves forward.

3.6. Processing a Data Indication

When the client receives a structurally valid Data indication, the client first checks whether the XOR-PEER-ADDRESS attribute value contains an IP address with which the client believes there is an active permission. If there is no such permission and the message is not a structurally valid ICE connectivity check, the client SHOULD discard the message. If the message is a structurally valid ICE connectivity check, the client parses, validates, and responds to the message as per standard ICE behavior.

4. ICE Interactions

The following subjects have been identified that should be discussed in greater detail:

o Interaction with ice-lite.

o Interactions with vanilla ice.

- Interactions with trickle ice.

In particular, this section should discuss setting IP address permissions as a result of receiving a valid ICE connectivity check and/or learning the true candidates via the answer.

5. Security Considerations

The following subjects have been identified that must be discussed in greater detail:

- An open port could be used to provide an unauthorized service. At this time, this is the primary security concern identified by the authors and some suitable mitigations should be discussed in this document.

- A valid ICE connectivity check could be replayed by an attacker. This risk is shared by existing address-based permissions and so is not a significant concern for this draft.

- An invalid ICE connectivity check using a snooped ufrag value could be forwarded to the client. This risk is also shared by existing address-based permissions and so is not a significant concern for this draft.

- Others?

6. IANA Considerations

[Paragraphs below in braces should be removed by the RFC Editor upon publication]

[The LOCAL-UFRAG attribute requires that IANA allocate a value in the "STUN attributes Registry" from the comprehension-required range (0x0000-0x7FFF), to be replaced for TBD throughout this document]

This document defines the LOCAL-UFRAG attribute, described in Section 3.1. IANA has allocated the comprehension-required codepoint TBD for this attribute.

7. Acknowledgements

Thanks to T. Reddy for early review of this draft.

8. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997,
<http://www.rfc-editor.org/info/rfc2119>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <http://www.rfc-editor.org/info/rfc5389>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <http://www.rfc-editor.org/info/rfc5766>.

[RFC6062] Perreault, S., Ed. and J. Rosenberg, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations", RFC 6062, DOI 10.17487/RFC6062, November 2010, <http://www.rfc-editor.org/info/rfc6062>.

Authors' Addresses

Brandon Williams
Akamai, Inc.
150 Broadway
Cambridge, MA 02142
USA

Email: brandon.williams@akamai.com

Justin Uberti
Google
747 6th Ave S
Kirkland, WA 98033
USA

Email: justin@uberti.name