TOC

MIP6	H. Tschofenig
Internet-Draft	Nokia Siemens Networks
Intended status: Standards Track	G. Bajko
Expires: August 28, 2008	Nokia
	February 25, 2008

Mobile IP Interactive Connectivity Establishment (M-ICE)
draft-tschofenig-mip6-ice-02.txt

Status of this Memo

Abstract

This document describes how the Interactive Connectivity Establishment (ICE) methodology can be used for Mobile IP to determine whether end-to-end communication is possible. ICE makes use of the Session Traversal Utilities for NAT (STUN) protocol in addition to mechanisms for checking connectivity between peers. After running the ICE the two MIP end points will be able to communicate directly or through a relay via Network Address Translators (NATs), Network Address and Port Translators (NAPTs) and firewalls.

This document addresses also the problems raised in RFC 4487 "Mobile IPv6 and Firewalls: Problem Statement".

Table of Contents

1. Introduction

In a typical Mobile IP deployment, there are two endpoints, mobile node and correspondent nodes, which want to communicate. They are able to communicate indirectly via a combination of Mobile IP signaling and reverse tunneling. A couple of different extensions are available for Mobile IP that allow multiple care-of addresses, IPv4/IPv6 interworking and different routes to be used through the network.
Unfortunately, it is likely that some of the available paths do not work due to connectivity problems caused by firewalling behavior. The VoIP community has investigated these problems extensively and developed a protocols and a methodology to systematically perform connectivity checks in order to determine a working path between the two end points. The Interactive Connectivity Establishment (ICE) specification describes how the Session Traversal Utilities for NAT (STUN) protocol can be used to execute these checks. This document suggests to utilize the ICE methodology and if possible STUN for Mobile IP, both Mobile IPv4 and Mobile IPv6. We call this usage Mobile IP - ICE, M-ICE for short.

> This document, however, concentrates on Mobile IPv6 as a starting point. A future version of this document will also describe the operation using Mobile IPv4. The ideal outcome is that the best available path through the network can be chosen when Mobile IP is used regardless of the MIP version and the environmental problems the two end points are facing.

At the beginning of the M-ICE process, the end points are ignorant of their own topologies. They might or might not be behind a NAT (or multiple tiers of NATs) and might be behind firewalls that limit the ability to communicate in different ways between the end points. M-ICE allows these end points to discover enough information about their topologies to potentially find one or more paths by which they can communicate.
Figure 1 (Overview) shows a typical environment for M-ICE deployment. The two end points are labelled L and R (for left and right, which helps visualize call flows). Both L and R are behind their own respective NATs or firewalls though they may not be aware of it. The type of NAT or firewall and their properties are also unknown. L and R are capable of engaging in an end-to-end mobility protocol exchange. This exchange will occur through mobility anchor points, such as Home Agents.
In this architecture the ICE functionality of TURN servers is provided by the Home Agent via reverse tunneling. In this document we assume that the STUN server is co-located with the Home Agent since it is convenient from a security and configuration point of view even though it is, from a solution point of view, not necessary.

```
           +--------+         Mobility           +--------+
           | Home   |         Signalling         | Home   |
           | Agent/ |----------------------------| Agent/ |
           | STUN   |                            | STUN   |
           | Server |                            | Server |
           +--------+                            +--------+
               ^                                     ^
               |                                     |
               |                                     |
     Mobility  |                                     |Mobility
     Signalling|                                     |Signalling
               |                                     |
               |                                     |
           +---v----+                            +---v----+
           | FW/NAT |                            | FW/NAT |
           +---^----+                            +---^----+
               |                                     |
               |                                     |
               v                                     v
           +--------+                            +--------+
           | Agent  |                            |  Agent |
           |   L    |                            |    R   |
           +--------+                            +--------+
```

Figure 1: Overview

The basic idea behind M-ICE is as follows: each end point has a variety of candidate TRANSPORT ADDRESSES (combination of IP address, transport protocol (UDP), and port) it could use to communicate with the other end point.

> To avoid unnecessary UDP encapsulation of end-to-end traffic in case there is no need todo so, it is also possible to consider using IP addresses rather than focusing exclusively on TRANSPORT ADDRESSES. For example, two MIP hosts behind the same NAT do not need to use UDP encapsulation. If there is no NAT or firewall between the two communicating nodes then there is again no need to provide support for UDP encapsulation. A future version of this document will provide support for this functionality.

Potentially, any of L's candidate transport addresses can be used to communicate with any of R's candidate transport addresses. In practice, however, many combinations do not work. For instance, if L and R are both behind NATs, their directly attached interface addresses (e.g., 192.168.1.100) are unlikely to be able to communicate. The purpose of M-ICE is to discover which pairs of addresses will work. The way that

M-ICE does this is to systematically try all possible pairs (in a carefully sorted order) until it finds one or more that works. Once found, the best pair is used for subsequent communication between the hosts.

1.1. Gathering Candidate Addresses

TOC

In order to execute ICE, an agent has to identify all of its address candidates. A CANDIDATE is a transport address - a combination of IP address and port for a particular transport protocol.
This document uses three types of candidates:

1. One viable candidate is a transport address obtained directly from a local interface. Such a candidate is called a HOST CANDIDATE.

2. Translated addresses on the public side of a NAT (called SERVER REFLEXIVE CANDIDATES). This address is obtained via STUN.

3. Addresses obtained via relaying traffic through a Home Agent, called RELAYED CANDIDATES.

1.2. Connectivity Checks

TOC

Once L has gathered all of its candidates, it orders them in highest to lowest priority and sends them to R over the signalling channel. We refer to the signaling channel to the end-to-end MIP exchange. The extension to exchange candidates can be found in Section 11 (Attribute Encoding).
When R receives the L's MIP message, R performs the same candidate gathering process and responds with its own list of candidates. At the end of this process, each agent has a complete list of both its candidates and its peer's candidates. It pairs them up, resulting in CANDIDATE PAIRS. To see which pairs work, each agent schedules a series of connectivity CHECKS. Each check is a STUN transaction that the client will perform on a particular candidate pair by sending a STUN request from the local candidate to the remote candidate; a response indicates there is connectivity to the peer using that candidate address.
It is important to note that the STUN requests are sent to and from the exact same IP addresses and ports that will be used for subsequent data traffic.

1.3. Sorting Candidates

TOC

Because the algorithm above searches all candidate pairs, if a working pair exists it will eventually find it no matter what order the candidates are tried in. In order to produce faster (and better) results, the candidates are sorted in a specified order. The resulting list of sorted candidate pairs is called the CHECK LIST.

1.4. Frozen Candidates

TOC

The concept of frozen candidates is not applied when ICE is applied to MIP. [Editor's Note: More investigations are needed to evaluate whether this is indeed true and the concept of frozen candidates can be ignored.]

1.5. Security for Checks

TOC

Because the ICE algorithm is used to discover which addresses can be used to send traffic between two end points, it is important to ensure that the process cannot be hijacked to send traffic to the wrong location. Each STUN connectivity check is covered by a message authentication code (MAC). There are two ways to generate the keying material for this MAC. Either keying material is derived from the keying material generated by the return routability procedure or new keying material is distributed separately as excercised in ICE.

> This document currently uses the latter technique without a strong preference.

In any case, this MAC provides message integrity and data origin authentication, thus stopping an attacker from forging or modifying connectivity check messages.

1.6. Concluding M-ICE

TOC

ICE checks are performed in a specific sequence, so that high priority candidate pairs are checked first, followed by lower priority ones.

2. Terminology

TOC

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 RFC 2119 [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).
This document heavily relies on the terminology introduced in [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.).

3. Design Choices

TOC

The work in this document is guided by the following design choices, namely:

*The offer/answer exchange described in ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) is mapped to the end-to-end MIP signaling exchange. For end-to-end communication this document assumes that MIPv6 signaling are allowed to be exchanged between the mobile node and the correspondent node. When Network Address Translators and firewalls are located along the path then direct end-to-end communication between the two end points is typically not possible and hence this protocol interaction is provided via MIP Home Agents. The functionality described in [I-D.bajko-mip6-rrtfw] (Bajko, G., “Firewall friendly Return-Routability Test (RRT) for Mobile IPv6,” February 2008.) is used.

*We assume that MIP initiators and MIP responders implement and use STUN. For performing connectivity checks a couple of other alternatives are, however, possible:

> It would be possible to utilize the SHIM6 REAchability Protocol (REAP) [I-D.ietf-shim6-failure-detection] (Arkko, J. and I. Beijnum, “Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming,” June 2008.) but STUN provides the same support with a more likely chance for widespread deployment. REAP currently only provides IPv6 support. It it obviously possible to turn a protocol in any other one.
>
> Custom MIP messages could be created.

- If one peer does not support STUN then the optimal results of M-ICE cannot be provided. There is, however, the ability to make use of STUN LITE when a host is on the public address space and known not to be behind a firewall.

- Obtaining Relay Addresses from STUN [I-D.ietf-behave-turn] (Rosenberg, J., Mahy, R., and P. Matthews, “Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN),” July 2009.), formerly known as TURN, is intentionally not used in this document. For MIP, the Home Agent tunneling functionality is used instead of TURN.

- This document makes use of the UDP-encapsulated of MIP packets, as specified in [I-D.ietf-mip6-nemo-v4traversal] (Soliman, H., “Mobile IPv6 support for dual stack Hosts and Routers (DSMIPv6),” November 2007.).

- This document focuses only on the data exchange between the two end points rather than on the communication between a mobile node and the Home Agent or on the ability to allow MIP signaling messages to traverse NATs and firewalls.

- Each STUN connectivity check is covered by a message authentication code (MAC) generated based on keying material derived from information carried in MIP messages, see Section 11 (Attribute Encoding). Alternatively, keying material could be derived from the return routability test procedure.

Note that the ICE description assumes usage within a VoIP environment where individual flows are controlled. However, the protocol interaction described in this document operates at a lower layer where application specific message flows are not visible. When a CANDIDATE PAIR, consisting of two TRANSPORT ADDRESSES, is created then it will typically refer to multiple flows then traffic between two end points experiences UDP encapsulation (due to the need to traverse a NAT or a stateful packet filtering firewall).
The descriptions in the ICE specification related to SIP, ANAT, RTP, RTCP, third party call control, preconditions, forking, etc. are not applicable to MIP and are not included in this document.

> From an editorial point of view it would be possible to copy-and-paste relevant parts of the ICE specification and to remove VoIP specific descriptions but for this version of the document we did not follow this approach.

The main accomplishment of this document is the reuse of the well-established ICE specification that builds on STUN. STUN enjoys widespread implementation support and maximum code re-use was one of the design criteria for this document.

4. Sending the Initial Offer

TOC

In order to send the initial offer in an offer/answer exchange, an agent must (1) gather candidates, (2) prioritize them, (3) choose default candidates, and then (4) formulate and send them to the other peer.
Section 4 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) is applicable to this document with the following two exceptions: First, TURN is not used in this document but instead similar functionality is accomplished via a Home Agent. Second, the description regarding encoding of candidates in SDP is not applicable and replaced by a MIP specific encoding described in Section 11 (Attribute Encoding).

5. Receiving the Initial Offer

TOC

When an agent receives an initial offer, it will check if the offerer supports sufficient ICE functionality to proceed (i.e., if both offerer and answerer are lite implementations, ICE cannot proceed), determine its own role, gather candidates, prioritize them, choose default candidates, encode and send an answer, and for full implementations, form the check lists and begin connectivity checks.
Again, the description regarding encoding of candidates in SDP is not applicable to this document and is replaced by a MIP specific encoding described in Section 11 (Attribute Encoding). Note that only the encoding is different but not the semantic. As such, the description in Section 5 of [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) is applicable to this document.

TOC

6. Receipt of the Initial Answer

Section 6 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) describes the procedures that an agent follows when it receives the answer from the peer. It verifies that its peer supports ICE, determines its role, and for full implementations, forms the check list and begins performing periodic checks.

7. Performing Connectivity Checks

TOC

Section 7 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) describes how connectivity checks are performed using STUN [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.) and the content of that section is fully applicable to this document.

8. Concluding ICE Processing

TOC

The description in Section 8 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) illustrates processing rules that apply only to full implementations. Concluding ICE involves nominating pairs by the controlling agent and updating of state machinery

9. Subsequent Offer/Answer Exchanges

TOC

Either agent may generate a subsequent offer at any time. The rules in Section 9 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) will cause the controlling agent to send an updated offer at the conclusion of ICE processing when ICE has selected different candidate pairs from the default pairs. Section 9 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/

Answer Protocols,” October 2007.) defines rules for construction of subsequent offers and answers.

Note that the term "media stream" in Section 9 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) translates to an individual UDP-encapsulated data flow exchanged between the two MIP end points.

10. Keepalives

TOC

Section 10 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) describes a keepalive mechanism. The RTP description, such as RTP No-Op and RTP comfort noise, is not applicable to this document. Other useful keepalive techniques are described in [I-D.marjou-behave-app-rtp-keepalive] (Marjou, X., “Application Mechanism for maintaining alive the Network Address Translator (NAT) mappings associated to RTP flows.,” February 2007.) and may be useful for MIP; a recommendation will be made in a subsequent version of this document.

11. Attribute Encoding

TOC

To accomplish the same functionality this specification needs to reuse the semantic, but not necessarily the encoding, of seven attributes defined in the ICE specification [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.), namely "candidate", "remote-candidates", "ice-lite", "ice-mismatch", "ice-ufrag", "ice-pwd" and "ice-options".

Section 15.1 to Section 15.5 of ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) describe the semantic of the attributes.

MIP-ICE Mobility Options Format:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type          |           Header Len          |# of candidates|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |L|M|         Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                       Ice-pwd                                 |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ice-ufrag                     |     ice-options (var)        ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                 Candidate 1                                   .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                 Candidate 2                                   .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                 Candidate n                                   .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where L: ice-lite
M: ice-mismatch
# of candidates: the number of candidates carried by this option

Ice-options:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length       | Data ...

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Candidate:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ver   |         Length                | type  |   comp-id     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    transport  |       Reserved                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Priority                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                     Connection-address                        .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    port                         |  rel-port                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                     Rel-address                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The fields have the following meaning:

ver: IP address version contained in Connection-address and
     Rel-address fields

type: cand-type as defined in ICE

comp-id: component-id as defined in ICE

transport: transport address

priority: sender priority assigned to the connection-address, as
          defined in ICE

connection-address: IP address, 32 bit if ver=4 and 128 bit if
                    ver=6

port: port number

rel-port: port number, as defined in ICE

rel-address: IP address, as defined in ICE
```

Figure 2: Attribute Encoding

12. Demultiplexing MIP and STUN messages

TOC

When MIP and STUN messages are run over the same port it is necessary to demultiplex them. For this usage it is necessary to have a FINGERPRINT attribute in place, as defined in [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P.,

and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.).
A STUN packet always has the fixed value 0x2112A442 in its Magic Cookie field (bits 32-64 from the beginning of the UDP payload).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0|     STUN Message Type     |         Message Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                              . . .
```

Figure 3: STUN Header

In this same offset from the UDP header, the MIP header has the Checksum field and the start of the Message Data field. The concatenation of the Checksum field and the first 16 bits of the Message Data field may coincide with the STUN Magic Cookie.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload Proto |  Header Len   |   MH Type     |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
.                                                               .
.                        Message Data                           .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 0                   1                   2                   3
                              . . .
```

Figure 4: MIP Header

When the value in that place equals the value of the STUN Magic Cookie, the presence of the STUN FINGERPRINT attribute tells unambigously whether this is a STUN message or not.
A future version will also discuss the demultiplexing when UDP encapsulation is not used.

13. Example

TOC

[Editor's Note: An example will be provided in the next version of the document.]

14. Security Considerations

TOC

There are several types of attacks possible in an M-ICE system. This section considers these attacks and their countermeasures.

14.1. Attacks on Connectivity Checks

TOC

An attacker might attempt to disrupt the STUN connectivity checks. Ultimately, all of these attacks fool an agent into thinking something incorrect about the results of the connectivity checks. The possible false conclusions an attacker can try and cause are:

> **False Invalid:**
> An attacker can fool a pair of agents into thinking a candidate pair is invalid, when it isn't. This can be used to cause an agent to prefer a different candidate (such as one injected by the attacker), or to disrupt a call by forcing all candidates to fail.
>
> **False Valid:**
> An attacker can fool a pair of agents into thinking a candidate pair is valid, when it isn't. This can cause an agent to proceed with a session, but then not be able to receive any data traffic.
>
> **False Peer-Reflexive Candidate:**
> An attacker can cause an agent to discover a new peer reflexive candidate, when it shouldn't have. This can be used to redirect data traffic to a DoS target or to

> the attacker, for eavesdropping or other purposes.
>
> **False Valid on False Candidate:**
> An attacker has already convinced an agent that there is a candidate with an address that doesn't actually route to that agent (for example, by injecting a false peer reflexive candidate or false server reflexive candidate). It must then launch an attack that forces the agents to believe that this candidate is valid.

Of the various techniques for creating faked STUN messages described in [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.), many are not applicable for the connectivity checks. Compromises of STUN servers are not much of a concern, since the STUN servers are embedded in endpoints and distributed throughout the network. Thus, compromising the peer's embedded STUN server is equivalent to compromising the end point, and if that happens, far more problematic attacks are possible than those against ICE.

Injection of fake responses and relaying modified requests all can be handled in ICE with the countermeasures discussed below.

To force the false invalid result, the attacker has to wait for the connectivity check from one of the agents to be sent. When it is, the attacker needs to inject a fake response with an unrecoverable error response, such as a 600. However, since the candidate is, in fact, valid, the original request may reach the peer agent, and result in a success response. The attacker needs to force this packet or its response to be dropped, through a DoS attack, layer 2 network disruption, or other technique. If it doesn't do this, the success response will also reach the originator, alerting it to a possible attack. Fortunately, this attack is mitigated completely through the STUN message integrity mechanism. The attacker needs to inject a fake response, and in order for this response to be processed, the attacker needs the password. If the candidates are exchange in MIP messages and therefore secured, the attacker will not have the password.

Forcing the fake valid result works in a similar way. The agent needs to wait for the Binding Request from each agent, and inject a fake success response. The attacker won't need to worry about disrupting the actual response since, if the candidate is not valid, it presumably wouldn't be received anyway. However, like the fake invalid attack, this attack is mitigated completely through the STUN message integrity and offer/answer security techniques.

Forcing the false peer reflexive candidate result can be done either with fake requests or responses, or with replays. We consider the fake requests and responses case first. It requires the attacker to send a Binding Request to one agent with a source IP address and port for the false candidate. In addition, the attacker must wait for a Binding Request from the other agent, and generate a fake response with a XOR-

MAPPED-ADDRESS attribute containing the false candidate. Like the other attacks described here, this attack is mitigated by the STUN message integrity mechanisms and secure offer/answer exchanges.

Forcing the false peer reflexive candidate result with packet replays is different. The attacker waits until one of the agents sends a check. It intercepts this request, and replays it towards the other agent with a faked source IP address. It must also prevent the original request from reaching the remote agent, either by launching a DoS attack to cause the packet to be dropped, or forcing it to be dropped using layer 2 mechanisms. The replayed packet is received at the other agent, and accepted, since the integrity check passes (the integrity check cannot and does not cover the source IP address and port). It is then responded to. This response will contain a XOR- MAPPED-ADDRESS with the false candidate, and will be sent to that false candidate. The attacker must then receive it and relay it towards the originator.

The other agent will then initiate a connectivity check towards that false candidate. This validation needs to succeed. This requires the attacker to force a false valid on a false candidate. Injecting of fake requests or responses to achieve this goal is prevented using the integrity mechanisms of STUN and the offer/answer exchange. Thus, this attack can only be launched through replays. To do that, the attacker must intercept the check towards this false candidate, and replay it towards the other agent. Then, it must intercept the response and replay that back as well.

This attack is very hard to launch unless the attacker is identified by the fake candidate. This is because it requires the attacker to intercept and replay packets sent by two different hosts. If both agents are on different networks (for example, across the public Internet), this attack can be hard to coordinate, since it needs to occur against two different endpoints on different parts of the network at the same time.

If the attacker them self is identified by the fake candidate the attack is easier to coordinate. However, since MIP utilizes IPsec ESP to protect the data traffic end-to-end, the attacker will not be able to inspect any application data, they will only be able to discard them. However, this attack requires the agent to disrupt packets in order to block the connectivity check from reaching the target. In that case, if the goal is to disrupt the end-to-end communication, its much easier to just disrupt it with the same mechanism, rather than attack ICE.

14.2. Attacks on Address Gathering

TOC

ICE endpoints make use of STUN for gathering candidates from a STUN server in the network. This is corresponds to the Binding Discovery usage of STUN described in [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.). As a consequence, the attacks against STUN itself that are described in that specification can still be used against the binding discovery usage when utilized with ICE.

However, the additional mechanisms provided by ICE actually counteract such attacks, making binding discovery with STUN more secure when combined with ICE.

Consider an attacker which is able to provide an agent with a faked mapped address in a STUN Binding Request that is used for address gathering. This is the primary attack primitive described in [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.). This address will be used as a server reflexive candidate in the ICE exchange. For this candidate to actually be used for media, the attacker must also attack the connectivity checks, and in particular, force a false valid on a false candidate. This attack is very hard to launch if the false address identifies a fourth party (neither the offerer, answerer, or attacker), since it requires attacking the checks generated by each agent in the session.

If the attacker elects not to attack the connectivity checks, the worst it can do is prevent the server reflexive candidate from being used. However, if the peer agent has at least one candidate that is reachable by the agent under attack, the STUN connectivity checks themselves will provide a peer reflexive candidate that can be used for the exchange of media. Peer reflexive candidates are generally preferred over server reflexive candidates. As such, an attack solely on the STUN address gathering will normally have no impact on a session at all.

14.3. Attacks on the Offer/Answer Exchanges

TOC

An attacker that can modify or disrupt the offer/answer exchanges themselves can readily launch a variety of attacks with M-ICE. They could direct data traffic to a target of a DoS attack, they could insert themselves into the data exchange, and so on. The security considerations of MIP apply.

14.4. Insider Attacks

TOC

In addition to attacks where the attacker is a third party trying to insert fake offers, answers or STUN messages, there are several attacks possible with ICE when the attacker is an authenticated and valid participant in the M-ICE exchange.

14.4.1. MIP Amplification Attack

TOC

In this attack, the attacker initiates communication to other agents, and maliciously includes the IP address and port of a DoS target as the destination for data traffic signaled in the MIP exchange.
This could causes substantial amplification; a single offer/answer exchange can create a continuing flood of data packets, possibly at high rates (consider video sources). This attack is not specific to ICE, but ICE can help provide remediation.
Specifically, if ICE is used, the agent receiving the malicious SDP will first perform connectivity checks to the target of media before sending media there. If this target is a third party host, the checks will not succeed, and media is never sent.
Unfortunately, ICE doesn't help if its not used, in which case an attacker could simply send the offer without the ICE parameters. However, in environments where the set of clients are known, and limited to ones that support ICE, the server can reject any offers or answers that don't indicate ICE support.

14.4.2. STUN Amplification Attack

TOC

The STUN amplification attack is similar to the MIP amplification attack. However, instead of data packets being directed to the target, STUN connectivity checks are directed to the target. The attacker sends an offer with a large number of candidates, say 50. The answerer receives the offer, and starts its checks, which are directed at the target, and consequently, never generate a response. The answerer will start a new connectivity check every 20ms, and each check is a STUN transaction consisting of 7 transmissions of a message 65 bytes in length (plus 28 bytes for the IP/UDP header) that runs for 7.9 seconds, for a total of 58 bytes/second per transaction on average. In the worst case, there can be 395 transactions in progress at once (7.9 seconds divided by 20ms), for a total of 182 kbps, just for STUN requests.
It is impossible to eliminate the amplification, but the volume can be reduced through a variety of heuristics. Agents SHOULD limit the total number of connectivity checks they perform to 100. Additionally, agents MAY limit the number of candidates they'll accept in an offer or answer.

15. IAB Considerations

TOC

The IAB has studied the problem of "Unilateral Self Address Fixing", which is the general process by which a agent attempts to determine its address in another realm on the other side of a NAT through a collaborative protocol reflection mechanism [RFC3424] (Daigle, L. and IAB, “IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation,” November 2002.). M-ICE is an example of a protocol that performs this type of function. Interestingly, the process for M-ICE is not unilateral, but bilateral, and the difference has a significant impact on the issues raised by IAB. M-ICE can be considered a B-SAF (Bilateral Self-Address Fixing) protocol, rather than an UNSAF protocol. Regardless, the IAB has mandated that any protocols developed for this purpose document a specific set of considerations. This section meets those requirements.

15.1. Problem Definition

TOC

From RFC 3424 [RFC3424] (Daigle, L. and IAB, “IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation,” November 2002.) any UNSAF proposal must provide:

Precise definition of a specific, limited-scope problem that is to be solved with the UNSAF proposal. A short term fix should not be generalized to solve other problems; this is why "short term fixes usually aren't".

The specific problems being solved by M-ICE are:

Provide a means for two peers to determine the set of transport addresses which can be used for communication.

Provide a means for resolving many of the limitations of other UNSAF mechanisms by wrapping them in an additional layer of processing (the M-ICE methodology).

Provide a means for a agent to determine an address that is reachable by another peer with which it wishes to communicate.

15.2. Exit Strategy

TOC

From RFC 3424, any UNSAF proposal must provide:

Description of an exit strategy/transition plan. The better short term fixes are the ones that will naturally see less and less use as the appropriate technology is deployed.

M-ICE itself doesn't easily get phased out. However, it is useful even in a globally connected Internet, to serve as a means for detecting whether communication paths are disrupted. M-ICE also helps prevent certain security attacks which have nothing to do with NAT. However, what M-ICE does is help phase out other UNSAF mechanisms. M-ICE effectively selects amongst those mechanisms, prioritizing ones that are better, and deprioritizing ones that are worse. Local IPv6 addresses can be preferred. As NATs begin to dissipate as IPv6 is introduced, server reflexive and relayed candidates (both forms of UNSAF mechanisms) simply never get used, because higher priority connectivity exists to the native host candidates. Therefore, the servers get used less and less, and can eventually be remove when their usage goes to zero.

Indeed, M-ICE can assist in the transition from IPv4 to IPv6. It can be used to determine whether to use IPv6 or IPv4 when two dual-stack hosts communicate. It can also allow a network with both 6to4 and native v6 connectivity to determine which address to use when communicating with a peer.

15.3. Brittleness Introduced by M-ICE

TOC

From RFC3424, any UNSAF proposal must provide:

Discussion of specific issues that may render systems more "brittle". For example, approaches that involve using data at multiple network layers create more dependencies, increase debugging challenges, and make it harder to transition.

M-ICE uses ICE that is utilizes [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.) instead of traditional STUN, RFC 3489 [RFC3489] (Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs),” March 2003.)). RFC 3489 has several points of brittleness. One of them is the discovery process which requires a agent to try and classify the type of NAT it is behind. This process is error-prone. With M-ICE, that discovery process is simply not used. Rather than unilaterally assessing the validity of the address, its validity is dynamically determined by measuring connectivity to a peer. The process of determining connectivity is very robust.

Another point of brittleness in traditional STUN is that it assumes that the STUN server is on the public Internet. Interestingly, with M-ICE, that is not necessary. There can be a multitude of STUN servers in a variety of address realms. ICE will discover the one that has provided a usable address.

The most troubling point of brittleness in traditional STUN is that it does not work in all network topologies. In cases where there is a shared NAT between each agent and the STUN server, traditional STUN may not work. With ICE, that restriction is removed.

Traditional STUN also introduces some security considerations. Fortunately, those security considerations are also mitigated by ICE.

Consequently, ICE serves to repair the brittleness introduced in other UNSAF mechanisms, and does not introduce any additional brittleness into the system.

With M-ICE Home Agents are used and they are assumed to be located on the public Internet to allow MIP to work.

15.4. Requirements for a Long Term Solution

TOC

From RFC 3424, any UNSAF proposal must provide:
Identify requirements for longer term, sound technical solutions -- contribute to the process of finding the right longer term solution.
M-ICE provides a long term solution by utilizing ICE concepts that have received a lot of peer review in the VoIP community and to apply them to MIP. The only other possible long term solutions are (a) to get rid of middleboxes, such as NATs and firewalls or to (b) interact with them. Regarding (b) extensions for STUN to allow the protocol to be deployed on NATs and firewalls is currently being investigated in [I-D.wing-behave-nat-control-stun-usage] (Wing, D., Rosenberg, J., and H. Tschofenig, “Discovering, Querying, and Controlling Firewalls and NATs,” October 2007.).

15.5. Issues with Existing NAPT Boxes

TOC

From RFC 3424, any UNSAF proposal must provide:
Discussion of the impact of the noted practical issues with existing, deployed NA[P]Ts and experience reports.
A number of NAT boxes are now being deployed into the market which try and provide "generic" ALG functionality. These generic ALGs hunt for IP addresses, either in text or binary form within a packet, and rewrite them if they match a binding. This interferes with traditional STUN. However, the update to STUN [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.) uses an encoding which hides these binary addresses from generic ALGs.
Existing NAPT boxes have non-deterministic and typically short expiration times for UDP-based bindings. This requires implementations to send periodic keepalives to maintain those bindings. ICE uses a default of 15s, which is a very conservative estimate. Eventually, over time, as NAT boxes become compliant to behave [RFC4787] (Audet, F. and C. Jennings, “Network Address Translation (NAT) Behavioral Requirements for Unicast UDP,” January 2007.), this minimum keepalive will become deterministic and well-known, and the ICE timers can be adjusted. Having a way to discover and control the minimum keepalive interval would be far better still.

16. Contributors

TOC

We would like to thank Thomas Schreck for his contributions to various aspects in this document.

17. Acknowledgments

TOC

The authors would like to thank Jonathan Rosenberg for his work on the ICE specification. This document copy-and-pastes text from the ICE specification. Hence, all the credits go to Jonathan.
Finally, Dan Wing and Philip Matthews helped us with the work on HIP-ICE.

18. References

TOC

18.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.
[I-D.ietf-mmusic-ice]	Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” draft-ietf-mmusic-ice-19 (work in progress), October 2007 (TXT).
[RFC3775]	Johnson, D., Perkins, C., and J. Arkko, “Mobility Support in IPv6,” RFC 3775, June 2004 (TXT).

18.2. Informative References

TOC

[I-D.ietf-behave-turn]	Rosenberg, J., Mahy, R., and P. Matthews, “Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN),” draft-ietf-behave-turn-16 (work in progress), July 2009 (TXT).
[I-D.ietf-shim6-failure-detection]	Arkko, J. and I. Beijnum, “Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming,” draft-ietf-shim6-failure-detection-13 (work in progress), June 2008 (TXT, PDF).

[RFC3489]	Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs),” RFC 3489, March 2003 (TXT).
[I-D.ietf-behave-rfc3489bis]	Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” draft-ietf-behave-rfc3489bis-18 (work in progress), July 2008 (TXT).
[RFC3424]	Daigle, L. and IAB, “IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation,” RFC 3424, November 2002 (TXT).
[I-D.wing-behave-nat-control-stun-usage]	Wing, D., Rosenberg, J., and H. Tschofenig, “Discovering, Querying, and Controlling Firewalls and NATs,” draft-wing-behave-nat-control-stun-usage-05 (work in progress), October 2007 (TXT).
[RFC4474]	Peterson, J. and C. Jennings, “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP),” RFC 4474, August 2006 (TXT).
[I-D.ietf-monami6-multiplecoa]	Wakikawa, R., Devarapalli, V., Tsirtsis, G., Ernst, T., and K. Nagami, “Multiple Care-of Addresses Registration,” draft-ietf-monami6-multiplecoa-14 (work in progress), May 2009 (TXT).
[I-D.ietf-mip6-nemo-v4traversal]	Soliman, H., “Mobile IPv6 support for dual stack Hosts and Routers (DSMIPv6),” draft-ietf-mip6-nemo-v4traversal-06 (work in progress), November 2007 (TXT).
[I-D.ietf-mip4-dsmipv4]	Tsirtsis, G., Park, V., and H. Soliman, “Dual Stack Mobile IPv4,” draft-ietf-mip4-dsmipv4-10 (work in progress), January 2009 (TXT).
[I-D.marjou-behave-app-rtp-keepalive]	Marjou, X., “Application Mechanism for maintaining alive the Network Address Translator (NAT) mappings associated to RTP flows.,” draft-marjou-behave-app-rtp-keepalive-01 (work in progress), February 2007 (TXT).
[RFC4787]	Audet, F. and C. Jennings, “Network Address Translation (NAT) Behavioral Requirements for Unicast UDP,” BCP 127, RFC 4787, January 2007 (TXT).
[I-D.bajko-mip6-rrtfw]	Bajko, G., “Firewall friendly Return-Routability Test (RRT) for Mobile IPv6,” draft-bajko-mip6-rrtfw-03 (work in progress), February 2008 (TXT).

Authors' Addresses

TOC

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland
Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@nsn.com
URI: http://www.tschofenig.com

Gabor Bajko
Nokia

Full Copyright Statement

TOC

Intellectual Property

that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.