BEHAVE
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2009

G. Camarillo
O. Novo
Ericsson
October 29, 2008

Traversal Using Relays around NAT (TURN) Extension for IPv4/IPv6 Transition
draft-ietf-behave-turn-ipv6-05.txt

Abstract

This document defines the REQUESTED-ADDRESS-TYPE attribute for Traversal Using Relays around NAT (TURN). The REQUESTED-ADDRESS-TYPE attribute allows a client to explicitly request the address type the TURN server will allocate (e.g., an IPv4-only node may request the TURN server to allocate an IPv6 address). Additionally, this document also defines a new error response code with the value 440 (Address Family not Supported).

Table of Contents

1. Introduction 3
2. Terminology 3
3. Overview of Operation 3
4. Creating an Allocation 4
4.1. Sending an Allocate Request 4
4.1.1. The REQUESTED-ADDRESS-TYPE Attribute 4
4.2. Receiving an Allocate Request 5
4.2.1. Unsupported Address Family 5
5. Refreshing an Allocation 6
5.1. Sending a Refresh Request 6
5.2. Receiving a Refresh Request 6
6. Packet Translations 6
7. Security Considerations 7
8. IANA Considerations 8
8.1. New STUN Attribute Registry 8
8.2. New STUN Response Code Registry 8
9. Acknowledgements 8
10. Normative References 8
Authors' Addresses 9
Intellectual Property and Copyright Statements 10

1. Introduction

Traversal Using Relays around NAT (TURN) [I-D.ietf-behave-turn] is a protocol that allows for an element behind a NAT or firewall to receive incoming data over UDP or TCP. It is most useful for elements behind symmetric NATs or firewalls that wish to be on the receiving end of a connection to a single peer.

This document defines the REQUESTED-ADDRESS-TYPE attribute, which is an extension to TURN that allows a client to explicitly request the address type the TURN server will allocate (e.g., an IPv4-only node may request the TURN server to allocate an IPv6 address).

This document also defines and registers a new error response code with the value 440 (Address Family not Supported).

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. Overview of Operation

When a user wishes a TURN server to allocate an address of a specific type, it sends an Allocate Request to the TURN server with a REQUESTED-ADDRESS-TYPE attribute. TURN can run over UDP and TCP, as it allows for a client to request address/port pairs for receiving both UDP and TCP.

Assuming the request is authenticated and has not been tampered with, the TURN server allocates a transport address of the type indicated in the REQUESTED-ADDRESS-TYPE attribute. This address is called the allocated transport address.

The TURN server returns the allocated address in the response to the Allocate Request. This response contains a RELAY-ADDRESS attribute indicating the mapped IP address and port that the server assigned to the client.

TURN servers allocate a single relayed-transport-address per allocation request. Therefore, Allocate Requests cannot carry more than one REQUESTED-ADDRESS-TYPE attribute. Consequently, a client that wishes to allocate more than one address at a TURN server (e.g., an IPv4 and an IPv6 address) needs to perform several allocation requests (one allocation request per address).

4. Creating an Allocation

The behavior specified here affects the processing defined in Section 6 of [I-D.ietf-behave-turn].

4.1. Sending an Allocate Request

A client that wishes to obtain a transport address of a specific address type includes a REQUESTED-ADDRESS-TYPE attribute in the Allocate Request that sends to the TURN server. Clients MUST NOT include more than one REQUESTED-ADDRESS-TYPE attribute in an Allocate Request. The mechanisms to formulate an Allocate Request are described in Section 6.1 of [I-D.ietf-behave-turn].

4.1.1. The REQUESTED-ADDRESS-TYPE Attribute

The REQUESTED-ADDRESS-TYPE attribute is used by clients to request the allocation of a specific address type from a server. The following is the format of the REQUESTED-ADDRESS-TYPE attribute. Note that TURN attributes are TLV (Type-Length-Value) encoded, with a 16 bit type, a 16 bit length, and a variable-length value.

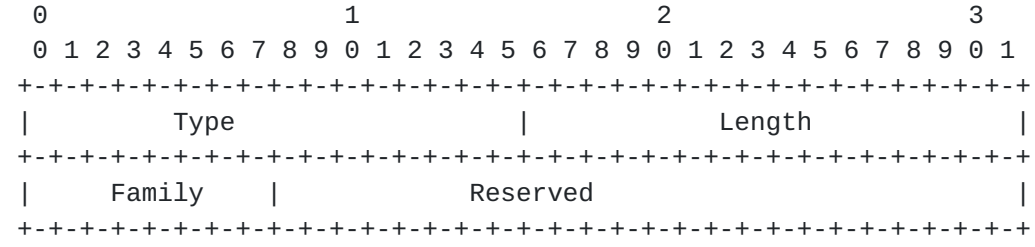

Figure 1: Format of REQUESTED-ADDRESS-TYPE Attribute

Type: the type of the REQUESTED-ADDRESS-TYPE attribute is 0x0017. As specified in [I-D.ietf-behave-rfc3489bis], attributes with values between 0x0000 and 0x7FFF are comprehension-required, which means that the client or server cannot successfully process the message unless it understands the attribute.

Length: this 16-bit field contains the length of the attribute in bytes. The length of this attribute is 4 bytes.

Family: there are two values defined for this field and specified in [I-D.ietf-behave-rfc3489bis]: 0x01 for IPv4 addresses and 0x02 for IPv6 addresses.

Reserved: at this point, the 24 bits in the reserved field SHOULD be set to zero by the client and MUST be ignored by the server.

The REQUEST-ADDRESS-TYPE attribute MAY only be present in Allocate Requests.

4.2. Receiving an Allocate Request

Assuming the request is authenticated and has not been tampered with, the TURN server processes the Allocate request. Following the rules in [I-D.ietf-behave-rfc3489bis], if the server does not understand the REQUESTED-ADDRESS-TYPE attribute, it generates an Allocate Error Response, which includes an ERROR-CODE attribute with response code 420 (Unknown Attribute). This response will contain an UNKNOWN-ATTRIBUTE attribute listing the unknown REQUESTED-ADDRESS-TYPE attribute.

If the server can successfully process the request, it allocates a transport address to the TURN client, called the allocated transport address, and returns it in the response to the Allocate Request.

As specified in [I-D.ietf-behave-turn], the Allocate Response contains the same transaction ID contained in the Allocate Request and the RELAY-ADDRESS attribute that sets it to the allocated transport address.

The RELAY-ADDRESS attribute indicates the mapped IP address and port. It is encoded in the same way as the XOR-MAPPED-ADDRESS [I-D.ietf-behave-rfc3489bis].

If the REQUESTED-ADDRESS-TYPE attribute is absent, the server MUST allocate an IPv4 transport address to the TURN client.

4.2.1. Unsupported Address Family

This document defines the following new error response code:

440 (Address Family not Supported): The server did not support the address family requested by the client. The client SHOULD NOT retry.

If the server does not support the address family requested by the client, it MUST generate an Allocate Error Response, and it MUST include an ERROR-CODE attribute with the response code defined in this draft, 440 (Address Family not Supported).

5. Refreshing an Allocation

The behavior specified here affects the processing defined in Section 7 of [I-D.ietf-behave-turn].

5.1. Sending a Refresh Request

To perform a binding refresh, the client generates a Refresh Request as described in Section 7.1 of [I-D.ietf-behave-turn]. The client MUST NOT include any REQUESTED-ADDRESS-TYPE attribute in its Refresh Request.

5.2. Receiving a Refresh Request

If a server receives a Refresh Request with a REQUESTED-ADDRESS-TYPE attribute, it MUST ignore the attribute and process the request as if the attribute was not there.

6. Packet Translations

The TURN specification [I-D.ietf-behave-turn] describes how TURN relays should relay traffic consisting of IPv4 packets (i.e., IPv4-to-IPv4 translations). The relay translates the IP addresses and port numbers of the packets based on the allocation's state data. How to translate other header fields is also specified in [I-D.ietf-behave-turn]. This document addresses IPv4-to-IPv6, IPv6-to-IPv4, and IPv6-to-IPv6 translations.

TURN relays performing any translation MUST translate the IP addresses and port numbers of the packets based on the allocation's state information as specified in [I-D.ietf-behave-turn]. TURN relays performing an IPv4-to-IPv6 or an IPv6-to-IPv4 translations SHOULD translate other header fields following SIIT (Stateless IP/ICMP Translation Algorithm) as described in Sections 3 and 4 of [RFC2765] respectively. Additionally, when the outgoing packet's size exceeds the outgoing link's MTU, the relay needs to generate an ICMP error (ICMPv6 Packet Too Big or ICMPv4 Destination Unreachable) reporting the MTU size. If the packet is being sent to the peer, the relay SHOULD reduce the MTU reported in the ICMP message by 48 bytes to allow room for the overhead of a Data indication.

> Note that the use of SIIT is at the "should" level. Having the use of SIIT at the "should" level instead of at the "must" level makes it possible to use different translation algorithms that may be developed in the future.

A TURN relay performing an IPv6-to-IPv6 translation translates other

header fields per the following rules:

Flow Label: the relay should consider that it is handling two different IPv6 flows. Therefore, the Flow label [RFC3697] SHOULD NOT be copied as part of the translation. The relay SHOULD set the Flow label to 0. The relay MAY choose to set the Flow label to a different value if it supports [RFC3697].

Hop Limit: the relay MUST act as a regular router with respect to decrementing the Hop Limit and generating an ICMPv6 error if it reaches zero.

Fragmentation: If the incoming packet did not include a Fragment header and the outgoing packet size does not exceed the outgoing link's MTU, the relay MUST send the outgoing packet without a Fragment header.

If the incoming packet did not include a Fragment header and the outgoing packet size exceeds the outgoing link's MTU, the delay MUST drop the outgoing packet and send an ICMP message of type 2 code 0 ("Packet too big") to the sender of the incoming packet. If the packet is being sent to the peer, the relay MUST reduce the MTU reported in the ICMP message by 48 bytes to allow room for the overhead of a Data indication.

If the incoming packet included a Fragment header and the outgoing packet size (with a Fragment header included) does not exceed the outgoing link's MTU, the relay MUST send the outgoing packet with a Fragment header. The relay MUST set the fields of the Fragment header as appropriate for a packet originating from the server.

If the incoming packet included a Fragment header and the outgoing packet size exceeds the outgoing link's MTU, the relay MUST fragment the outgoing packet into fragments of no more than 1280 bytes. The relay MUST set the fields of the Fragment header as appropriate for a packet originating from the server.

Extension Headers: the relay SHOULD send outgoing packet without any IPv6 extension headers, with the exception of the Fragmentation header as described above.

7. Security Considerations

The attribute and error response code defined in this document do not have any special security considerations beyond those for other attributes and Error response codes. All the security considerations applicable to STUN [I-D.ietf-behave-rfc3489bis] and TURN are

applicable to this document as well.

8. IANA Considerations

The IANA is requested to register the following values under the STUN Attributes registry and under the STUN Response Code Registry.

8.1. New STUN Attribute Registry

0x0017: REQUESTED-ADDRESS-TYPE

8.2. New STUN Response Code Registry

440 Address Family not Supported

9. Acknowledgements

The authors would like to thank Alfred E. Heggestad, Remi Denis-Courmont, and Philip Matthews for their feedback on this document.

10. Normative References

[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.

[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-09 (work in progress), July 2008.

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

[RFC2765] Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC 2765, February 2000.

[RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, March 2004.

Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

Oscar Novo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Oscar.Novo@ericsson.com

Full Copyright Statement

Intellectual Property