HIP Working Group
Internet-Draft
Intended status: Experimental
Expires: October 11, 2010

A. Keranen
J. Melen
Ericsson
April 9, 2010

Native NAT Traversal Mode for the Host Identity Protocol
draft-keranen-hip-native-nat-traversal-01

Abstract

This document specifies a new Network Address Translator (NAT) traversal mode for the Host Identity Protocol (HIP). The new mode is based on the Interactive Connectivity Establishment (ICE) methodology and UDP encapsulation of data and signaling traffic. The main difference from the previously specified modes is the use of HIP messages for all NAT traversal procedures.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 11, 2010.

Copyright Notice

Table of Contents

1. Introduction

The Host Identity Protocol (HIP) [RFC5201] is specified to run directly on top of IPv4 or IPv6. However, many middleboxes found in the Internet, such as NATs and firewalls, often allow only UDP or TCP traffic to pass [RFC5207]. Also, especially NATs usually require the host behind a NAT to create a forwarding state in the NAT before other hosts outside of the NAT can contact the host behind the NAT. To overcome this problem, different methods, commonly referred to as NAT traversal techniques, have been developed.

Two NAT traversal techniques for HIP are specified in [I-D.ietf-hip-nat-traversal]. One of them uses only UDP encapsulation, while the other uses also the Interactive Connectivity Establishment (ICE) [I-D.ietf-mmusic-ice] protocol, which in turn uses Session Traversal Utilities for NAT (STUN) [RFC5389] and Traversal Using Relays around NAT (TURN) [I-D.ietf-behave-turn] protocols to achieve a reliable NAT traversal solution.

The benefit of using ICE and STUN/TURN is that one can re-use the NAT traversal infrastructure already available in the Internet, such as STUN and TURN servers. Also, some middleboxes may be STUN-aware and could be able to do something "smart" when they see STUN being used for NAT traversal. However, implementing a full ICE/STUN/TURN protocol stack results in a considerable amount of effort and code which could be avoided by re-using and extending HIP messages and state machines for the same purpose. Thus, this document specifies a new NAT traversal mode that uses HIP messages instead of STUN for the connectivity checks, keepalives, and data relaying.

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 RFC 2119 [RFC2119].

This document uses the same terminology as [I-D.ietf-hip-nat-traversal] and the following:

HIP data relay:
: A host that forwards HIP data packets, such as Encapsulating Security Payload (ESP) [RFC5202], between two hosts.

Registered host:
A host that has registered for a relaying service with a HIP data relay.

3. Protocol Description

This section describes the normative behavior of the protocol extension. Most of the procedures are similar to what is defined in [I-D.ietf-hip-nat-traversal] but with different, or additional, parameter types and values. In addition, a new type of relaying server, HIP data relay, is specified.

3.1. Relay Registration

Relay registration procedure for HIP signaling is identical to the one specified in Section 4.1 of [I-D.ietf-hip-nat-traversal]. However, a host MAY also register for UDP encapsulated ESP relaying using Registration Type value RELAY_UDP_ESP (value 3).

If the HIP relay server supports relaying of UDP encapsulated ESP, the host is allowed to register for data relaying service (see Section 3.2), and the relay has relaying resources (free port numbers, bandwidth, etc.) available, the relay opens a UDP port on one of its addresses and signals the address and port to the registering host using the RELAYED_ADDRESS parameter (see Section 4.1 for details). If the relay would accept the data relaying request but does not have enough resources to provide data relaying service, it MUST reject the request with Failure Type 2 (Insufficient resources).

The registered host MUST maintain an active HIP association with the data relay as long as it requires the data relaying service. When the HIP association is closed (or times out), or the registration lifetime passes without the registered host refreshing the registration, the data relay MUST stop relaying packets for that host and close the corresponding UDP port.

The data relay MAY use the same relayed address and port for multiple registered hosts, but since this can cause problems with stateful firewalls (see Section 5) it is NOT RECOMMENDED.

3.2. Registration Authentication

If the HIP data relay knows the Host Identities (HIs) of all the hosts that are allowed to use the relaying service, it SHOULD reject registrations from unknown hosts. However, since it may be

unfeasible to pre-configure the relay with all the HIs, the relay SHOULD also support HIP certificates [I-D.ietf-hip-cert] to allow for certificate based authentication.

When a host wants to register with a HIP data relay, it SHOULD check if it has a suitable certificate for authenticating with the relay. How the suitability is determined and how the certificates are obtained is out of scope for this document. If the host has one or more suitable certificates, the host SHOULD include them (or just the most suitable one) in a CERT parameter to the HIP packet along with the REG_REQUEST parameter. If the host does not have any suitable certificates, it SHOULD send the registration request without the CERT parameter to test whether the relay accepts the request based on the host's identity.

When a relay receives a HIP packet with a REG_REQUEST parameter, and it requires authentication for at least one of the Registration Types listed in the REG_REQUEST parameter, it MUST first check whether the HI of the registering host is in the allowed list for all the Registration Types in the REG_REQUEST parameter. If the host is in the allowed list (or the relay does not require any authentication), the relay MUST proceed with the registration.

If the host was not in the allowed list and the relay requires hosts to authenticate, the relay MUST check whether the packet also contains a CERT parameter. If the packet does not contain a CERT parameter, the server MUST reject the registrations requiring authentication with Failure Type 0 (Registration requires additional credentials) [RFC5203]. If the certificate is valid and accepted (issued for the registering host and signed by a trusted issuer), the relay MUST proceed with the registration. If the certificate in the parameter is not accepted, the relay MUST reject the corresponding registrations with Failure Type 3 (Invalid certificate).

3.3. Forwarding Rules and Permissions

The HIP data relay uses a similar permission model as a TURN server: before any ESP data packets sent by a peer are forwarded, a permission must be set for the peer's address. The permissions also install a forwarding rule, similar to TURN's channels, based on the Security Parameter Index (SPI) values in the ESP packets.

Permissions are not required for the connectivity checks, but if a relayed address is selected to be used for data, the registered host MUST send an UPDATE message with a PEER_PERMISSION parameter with the address of the peer and the outbound and inbound SPI values the host is using with this peer.

When a data relay receives an UPDATE with a PEER_PERMISSION parameter, it MUST check if the sender of the UPDATE is registered for data relaying service, and drop the UPDATE if the host was not registered. If the host was registered, the relay checks if there is a permission with matching information (address, protocol, port and SPI values). If there is no such permission, a new permission is created and its lifetime is set to 5 minutes. If an identical permission already existed, it is refreshed by setting the lifetime to 5 minutes. A registered host SHOULD refresh permissions roughly 1 minute before the expiration if the permission is still needed.

3.4. Relaying UDP Encapsulated Data and Control Packets

When a HIP data relay accepts to relay UDP encapsulated data, it opens a UDP port (relayed address) for this purpose as described in Section 3.1. If the data relay receives a UDP encapsulated HIP control packet on that port, it MUST forward the packet to the registered host and add a RELAY_FROM parameter to the packet as if the data relay was acting as a HIP relay server [I-D.ietf-hip-nat-traversal].

When a host wants to send a HIP control packet (such as a connectivity check packet) to a peer via the data relay, it MUST add a RELAY_TO parameter containing the peer's address to the packet and send it to the data relay's address. The data relay MUST send the packet to the peer's address from the relayed address.

If the data relay receives a UDP packet that is not a HIP control packet to the relayed address, it MUST check whether there is a permission set for the peer the packet is coming from (i.e., the sender's address and SPI value matches to an installed permission), and if there is, it MUST forward the packet to the registered host that created the permission. Packets without a permission MUST be dropped silently.

When a host wants to send a UDP encapsulated ESP packet to a peer via the data relay, it MUST have an active permission at the data relay for the peer with the outbound SPI value it is using. The host MUST send the UDP encapsulated ESP packet to the data relay's address.

When the data relay receives a UDP encapsulated ESP packet from a registered host, it MUST check whether there exists a permission for that outbound SPI value. If such permission exists, the packet MUST be forwarded to the address that was registered for the SPI value. If no permission exists, the packet is dropped.

3.5. Candidate Gathering

A host needs to gather a set of address candidates before starting the connectivity checks. One server reflexive candidate can be discovered during the registration with the HIP relay server from the REG_FROM parameter.

If a host has more than one network interface, additional server reflexive candidates can be discovered by sending registration requests with Registration Type CANDIDATE_DISCOVERY (value 4) from each of the interfaces to a HIP relay server. When a HIP relay server receives a registration request with CANDIDATE_DISCOVERY type, it MUST add a REG_FROM parameter, containing the same information as if this was a relay registration, to the response. This request type SHOULD NOT create any state at the HIP relay server.

It is RECOMMENDED that the host also obtains a relayed candidate from a HIP data relay as described in Section 3.1.

Gathering of candidates MAY also be performed like specified in Section 4.2 of [I-D.ietf-hip-nat-traversal] if STUN and TURN servers are available, or if the host has just a single interface and there are no TURN or HIP data relay servers available.

3.6. Base Exchange via HIP Relay Server

The Base Exchange is performed as described in Section 4.5 of [I-D.ietf-hip-nat-traversal], except that "ICE candidates" are replaced by the candidates gathered using procedures described in Section 3.5

3.7. Native NAT Traversal Mode Negotiation

A host implementing this specification can signal the support for the native HIP NAT traversal mode by adding ICE-HIP-UDP NAT traversal mode (value 3) in the NAT_TRAVERSAL_MODE [I-D.ietf-hip-nat-traversal] parameter. If this mode is supported by both endpoints, and is the most preferred mode out of the all supported modes, further NAT traversal procedures are performed as specified in this document.

3.8. Connectivity Check Pacing Negotiation

Since the NAT traversal mode specified in this document utilizes connectivity checks, the check pacing negotiation MUST be performed as specified in Section 4.4 of [I-D.ietf-hip-nat-traversal]. New connectivity check transactions MUST NOT be started faster than once every Ta (the value negotiated with the TRANSACTION_PACING parameter).

3.9. Connectivity Checks

The connectivity checks are performed as described in Section 4.6 of [I-D.ietf-hip-nat-traversal] but instead of STUN packets, the connectivity checks are HIP UPDATE packets. See Section 4.3 for parameter details.

As defined in [I-D.ietf-hip-nat-traversal], both hosts MUST form a priority ordered checklist and start check transactions every Ta milliseconds as long as the checks are running and there are candidate pairs whose tests have not started. The retransmission timeout (RTO) for the connectivity check UPDATE packets MUST be calculated as defined in Section 4.6 of [I-D.ietf-hip-nat-traversal].

All connectivity check request packets MUST contain a CANDIDATE_PRIORITY parameter with the priority value that would be assigned to a peer reflexive candidate if one was learned from this check. The UPDATE packets that acknowledge a connectivity check requests MUST be sent from the same address that received the check and to the same address where the check was received from.

The acknowledgment UPDATE packets MUST contain a MAPPED_ADDRESS parameter with the port, protocol, and IP address of the address where the connectivity check request was received from.

After a working candidate pair, or pairs, have been discovered, the controlling host MUST conclude the checks by nominating the highest priority candidate pair for use. The pair MUST be nominated by sending an ESP packet on the selected pair. If the controlling host does not have any data to send, it SHOULD send an ICMP echo request using the nominated pair to signal to the controlled host that it can stop checks and start using the nominated pair.

If the connectivity checks failed the hosts SHOULD notify each other about the failure with a CONNECTIVITY_CHECKS_FAILED NOTIFY packet.

3.10. NAT Keepalives

To keep the NAT bindings towards the HIP relay server and the HIP data relay alive, if a registered host has not sent any data or control messages to the relay for 15 seconds, it MUST send a HIP NOTIFY packet to the relay. Likewise, if the host has not sent any data to a host it has security association and has run connectivity checks with, it MUST send either a HIP NOTIFY packet or an ICMP echo request using the same locators as the security association is using.

3.11. Handling Conflicting SPI Values

Since the HIP data relay determines from the SPI value to which peer an ESP packet should be forwarded, the outbound SPI values need to be unique for each relayed address registration. Thus, if a registered host detects that a peer would use an SPI value that is already used with another peer via the relay, it MUST NOT select the relayed address for use. The host MAY restart the base exchange to avoid a conflict or it MAY refrain from using the relayed candidate for the connectivity checks.

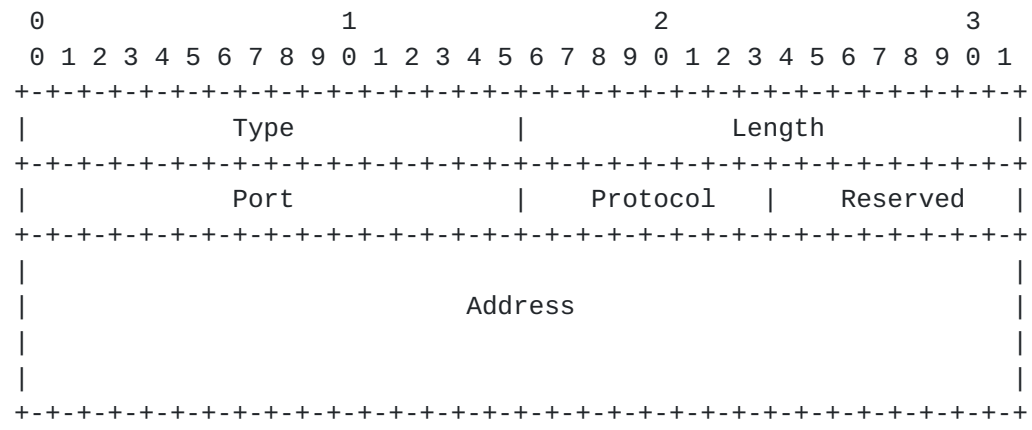
Since the SPI space is 32 bits and the SPI values should be random, the probability for a conflicting SPI value is fairly small. However, a host with many peers MAY decrease the odds of a conflict by registering more than one relayed address using different local addresses.

4. Packet Formats

The following subsections define the parameter and packet encodings for the new HIP parameters used for NAT traversal. UDP encapsulation of the HIP and ESP packets and format of the other required parameters is specified in Section 5 of [I-D.ietf-hip-nat-traversal].

4.1. RELAYED_ADDRESS and MAPPED_ADDRESS Parameters

The format of the RELAYED_ADDRESS and MAPPED_ADDRESS parameters (Figure 1) is identical to REG_FROM, RELAY_FROM and RELAY_TO parameters. This document specifies only use of UDP relaying and thus only protocol 17 is allowed. However, future documents may specify support for other protocols.

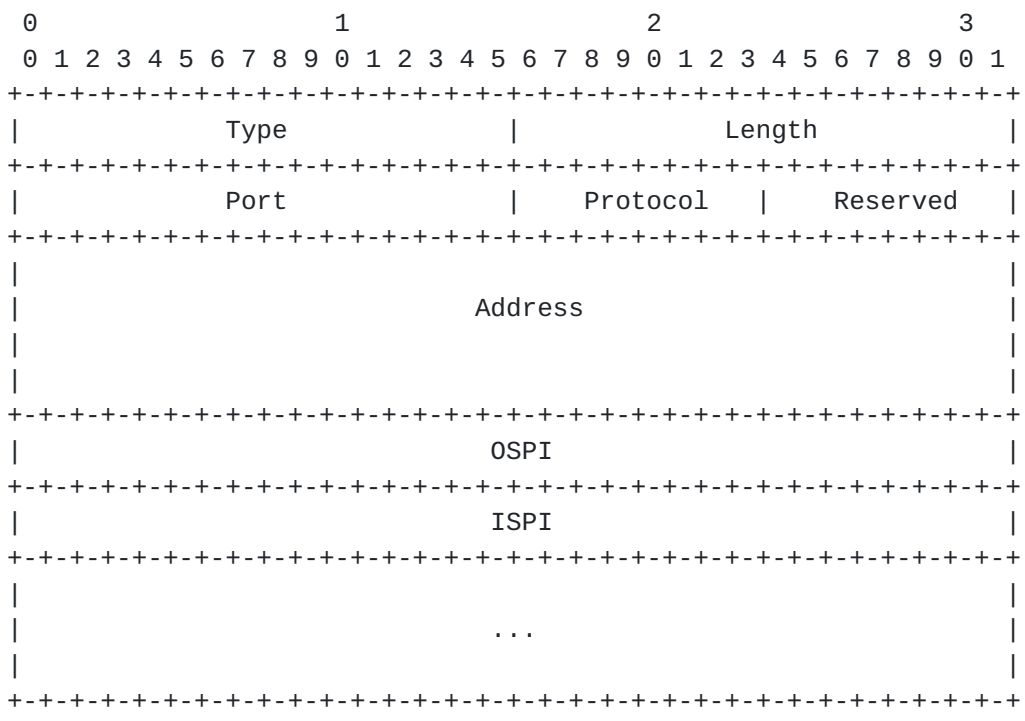


Type	[TBD by IANA; 952]
Length	20
Port	the UDP port number
Protocol	IANA assigned, Internet Protocol number (17 for UDP)
Reserved	reserved for future use; zero when sent, ignored when received
Address	an IPv6 address or an IPv4 address in "IPv4-Mapped IPv6 address" format

Figure 1: Format of the RELAYED_ADDRESS and MAPPED_ADDRESS Parameters

4.2. PEER_PERMISSION Parameter

The format of the PEER_PERMISSION parameter is shown in Figure 2. The parameter is used for setting up and refreshing forwarding rules and permissions at the data relay for data packets. The parameter contains one or more sets of Port, Protocol, Address, Outbound SPI, and Inbound SPI values. One set defines a rule for one peer address.

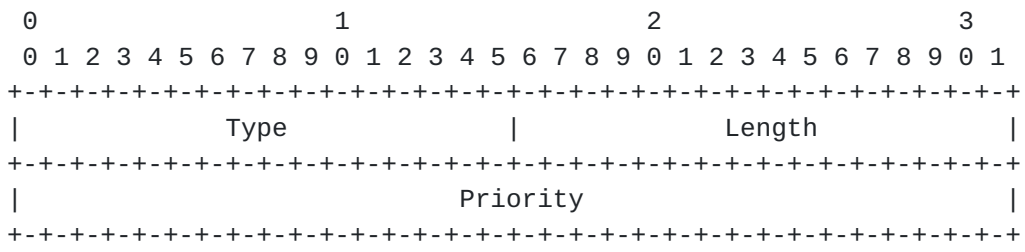


Type	[TBD by IANA; 1020]
Length	length in octets, excluding Type and Length
Port	the transport layer (UDP) port number
Protocol	IANA assigned, Internet Protocol number (17 for UDP)
Reserved	reserved for future use; zero when sent, ignored when received
Address	an IPv6 address, or an IPv4 address in "IPv4-Mapped IPv6 address" format, of the peer
OSPI	the outbound SPI value the registered host is using for the peer with the Address and Port
ISPI	the inbound SPI value the registered host is using for the peer with the Address and Port

Figure 2: Format of the PEER_PERMISSION Parameter

4.3. HIP Connectivity Check Packets

The connectivity request messages are HIP UPDATE packets with CANDIDATE_PRIORITY parameter (Figure 3). Response UPDATE packets contain a MAPPED_ADDRESS parameter (Figure 1).



Type [TBD by IANA; 954]
Length 4
Priority the priority of a peer reflexive candidate

Figure 3: Format of the CANDIDATE_PRIORITY Parameter

5. Security Considerations

If the data relay uses the same relayed address and port for multiple registered hosts, it appears to all the peers, and their firewalls, that all the registered hosts using the relay are at the same address. Thus, a stateful firewall may allow packets pass from hosts that would not normally be able to send packets to a peer behind the firewall. Therefore, a HIP data relay SHOULD NOT re-use the port numbers. If port numbers need to be re-used, the relay SHOULD have a sufficiently large pool of port numbers and select ports from the pool randomly to decrease the chances of a registered host obtaining the same address that a certain other host is using.

6. Acknowledgements

This document re-uses many of the ideas proposed in various earlier HIP NAT traversal related drafts by Miika Komu, Simon Schuetz, Martin Stiemerling, Pekka Nikander, Marcelo Bagnulo, Vivien Schmitt, Abhinav Pathak, Lars Eggert, Thomas Henderson, Hannes Tschofenig, and Philip Matthews.

7. IANA Considerations

This section is to be interpreted according to [RFC5226].

This document updates the IANA Registry for HIP Parameter Types [RFC5201] by assigning new HIP Parameter Type value for the new HIP Parameter: RELAYED_ADDRESS (defined in Section 4.1).

This document also updates the IANA Registry for HIP NAT traversal modes [I-D.ietf-hip-nat-traversal] by assigning value for the NAT

traversal mode ICE-HIP-UDP (defined in Section 3.7).

This document defines additional registration types for the HIP Registration Extension [RFC5203] that allow registering with a HIP relay server for ESP relaying service: RELAY_UDP_ESP (defined in Section 3.1); and performing server reflexive candidate discovery: CANDIDATE_DISCOVERY (defined in Section 3.5).

The IANA Registry for HIP Registration Failure Types is updated with new Failure Types "Insufficient resources" (defined in Section 3.1) and "Invalid certificate" (defined in Section 3.2).

8. References

8.1. Normative References

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

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5202] Jokela, P., Moskowitz, R., and P. Nikander, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)", RFC 5202, April 2008.

[RFC5203] Laganier, J., Koponen, T., and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", RFC 5203, April 2008.

[RFC5207] Stiemerling, M., Quittek, J., and L. Eggert, "NAT and Firewall Traversal Issues of Host Identity Protocol (HIP) Communication", RFC 5207, April 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

[I-D.ietf-hip-nat-traversal] Komu, M., Henderson, T., Tschofenig, H., Melen, J., and A. Keranen, "Basic HIP Extensions for Traversal of Network Address Translators", draft-ietf-hip-nat-traversal-09 (work in progress), October 2009.

[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.

[I-D.ietf-hip-cert]
Heer, T. and S. Varjonen, "HIP Certificates", draft-ietf-hip-cert-02 (work in progress), October 2009.

8.2. Informative References

[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.

Authors' Addresses

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

Email: Ari.Keranen@ericsson.com

Jan Melen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

Email: Jan.Melen@ericsson.com