Network Working Group
Request for Comments: 5204
Category: Experimental

J. Laganier
DoCoMo Euro-Labs
L. Eggert
Nokia

April 2008

Host Identity Protocol (HIP) Rendezvous Extension

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Abstract

This document defines a rendezvous extension for the Host Identity Protocol (HIP). The rendezvous extension extends HIP and the HIP registration extension for initiating communication between HIP nodes via HIP rendezvous servers. Rendezvous servers improve reachability and operation when HIP nodes are multi-homed or mobile.

Table of Contents

1. Introduction

The Host Identity Protocol (HIP) Architecture [RFC4423] introduces the rendezvous mechanism to help a HIP node to contact a frequently moving HIP node. The rendezvous mechanism involves a third party, the rendezvous server (RVS), which serves as an initial contact point ("rendezvous point") for its clients. The clients of an RVS are HIP nodes that use the HIP Registration Extension [RFC5203] to register their HIT->IP address mappings with the RVS. After this registration, other HIP nodes can initiate a base exchange using the IP address of the RVS instead of the current IP address of the node they attempt to contact. Essentially, the clients of an RVS become reachable at the RVS's IP address. Peers can initiate a HIP base exchange with the IP address of the RVS, which will relay this initial communication such that the base exchange may successfully complete.

2. Terminology

This section defines terms used throughout the remainder of this specification.

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].

In addition to the terminology defined in the HIP specification [RFC5201] and the HIP Registration Extension [RFC5203], this document defines and uses the following terms:

Rendezvous Service
: A HIP service provided by a rendezvous server to its rendezvous clients. The rendezvous server offers to relay some of the arriving base exchange packets between the initiator and responder.

Rendezvous Server (RVS)
: A HIP registrar providing rendezvous service.

Rendezvous Client
: A HIP requester that has registered for rendezvous service at a rendezvous server.

Rendezvous Registration
: A HIP registration for rendezvous service, established between a rendezvous server and a rendezvous client.

3. Overview of Rendezvous Server Operation

Figure 1 shows a simple HIP base exchange without a rendezvous server, in which the initiator initiates the exchange directly with the responder by sending an I1 packet to the responder's IP address, as per the HIP specification [RFC5201].

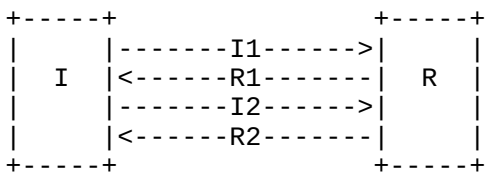


Figure 1: HIP base exchange without rendezvous server.

The End-Host Mobility and Multihoming with the Host Identity Protocol specification [RFC5206] allows a HIP node to notify its peers about changes in its set of IP addresses. This specification presumes initial reachability of the two nodes with respect to each other.

However, such a HIP node MAY also want to be reachable to other future correspondent peers that are unaware of its location change. The HIP Architecture [RFC4423] introduces rendezvous servers with whom a HIP node MAY register its host identity tags (HITs) and current IP addresses. An RVS relays HIP packets arriving for these HITs to the node's registered IP addresses. When a HIP node has registered with an RVS, it SHOULD record the IP address of its RVS in its DNS record, using the HIP DNS resource record type defined in the HIP DNS Extension [RFC5205].

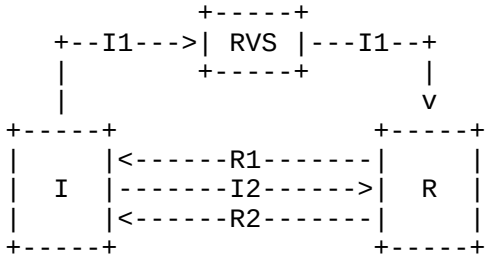


Figure 2: HIP base exchange with a rendezvous server.

Figure 2 shows a HIP base exchange involving a rendezvous server. It is assumed that HIP node R previously registered its HITs and current IP addresses with the RVS, using the HIP Registration Extension [RFC5203]. When the initiator I tries to establish contact with the

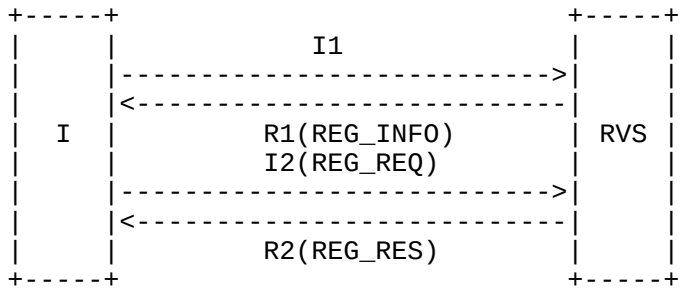
responder R, it must send the I1 of the base exchange either to one of R's IP addresses (if known via DNS or other means) or to one of R's rendezvous servers. Here, I obtains the IP address of R's rendezvous server from R's DNS record and then sends the I1 packet of the HIP base exchange to RVS. RVS, noticing that the HIT contained in the arriving I1 packet is not one of its own, MUST check its current registrations to determine if it needs to relay the packets. Here, it determines that the HIT belongs to R and then relays the I1 packet to the registered IP address. R then completes the base exchange without further assistance from RVS by sending an R1 directly to the I's IP address, as obtained from the I1 packet. In this specification, the client of the RVS is always the responder. However, there might be reasons to allow a client to initiate a base exchange through its own RVS, like NAT and firewall traversal. This specification does not address such scenarios, which should be specified in other documents.

3.1. Diagram Notation

Notation	Significance
I, R	I and R are the respective source and destination IP addresses in the IP header.
HIT-I, HIT-R	HIT-I and HIT-R are the initiator's and the responder's HITs in the packet, respectively.
REG_REQ	A REG_REQUEST parameter is present in the HIP header.
REG_RES	A REG_RESPONSE parameter is present in the HIP header.
FROM:I	A FROM parameter containing the IP address I is present in the HIP header.
RVS_HMAC	An RVS_HMAC parameter containing an HMAC keyed with the appropriate registration key is present in the HIP header.
VIA:RVS	A VIA_RVS parameter containing the IP address RVS of a rendezvous server is present in the HIP header.

3.2. Rendezvous Client Registration

Before a rendezvous server starts to relay HIP packets to a rendezvous client, the rendezvous client needs to register with it to receive rendezvous service by using the HIP Registration Extension [RFC5203] as illustrated in the following schema:

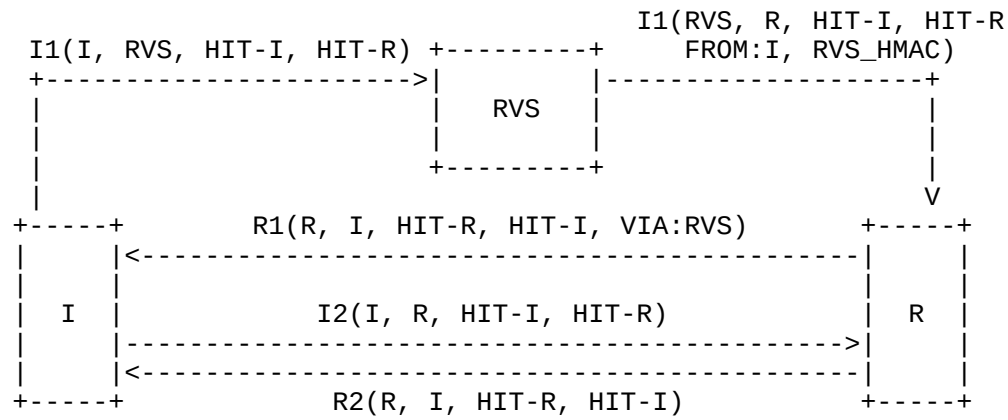


Rendezvous client registering with a rendezvous server.

3.3. Relaying the Base Exchange

If a HIP node and one of its rendezvous servers have a rendezvous registration, the rendezvous servers relay inbound I1 packets (that contain one of the client's HITs) by rewriting the IP header. They replace the destination IP address of the I1 packet with one of the IP addresses of the owner of the HIT, i.e., the rendezvous client. They MUST also recompute the IP checksum accordingly.

Because of egress filtering on the path from the RVS to the client [RFC2827][RFC3013], a HIP rendezvous server SHOULD replace the source IP address, i.e., the IP address of I, with one of its own IP addresses. The replacement IP address SHOULD be chosen according to relevant IPv4 and IPv6 specifications [RFC1122][RFC3484]. Because this replacement conceals the initiator's IP address, the RVS MUST append a FROM parameter containing the original source IP address of the packet. This FROM parameter MUST be integrity protected by an RVS_HMAC keyed with the corresponding rendezvous registration integrity key [RFC5203].



Rendezvous server rewriting IP addresses.

This modification of HIP packets at a rendezvous server can be problematic because the HIP protocol uses integrity checks. Because the I1 does not include HMAC or SIGNATURE parameters, these two end-to-end integrity checks are unaffected by the operation of rendezvous servers.

The RVS SHOULD verify the checksum field of an I1 packet before doing any modifications. After modification, it MUST recompute the checksum field using the updated HIP header, which possibly included new FROM and RVS_HMAC parameters, and a pseudo-header containing the updated source and destination IP addresses. This enables the responder to validate the checksum of the I1 packet "as is", without having to parse any FROM parameters.

4. Rendezvous Server Extensions

This section describes extensions to the HIP Registration Extension [RFC5203], allowing a HIP node to register with a rendezvous server for rendezvous service and notify the RVS aware of changes to its current location. It also describes an extension to the HIP specification [RFC5201] itself, allowing establishment of HIP associations via one or more HIP rendezvous server(s).

4.1. RENDEZVOUS Registration Type

This specification defines an additional registration for the HIP Registration Extension [RFC5203] that allows registering with a rendezvous server for rendezvous service.

Number	Registration Type
1	RENDEZVOUS

4.2. Parameter Formats and Processing

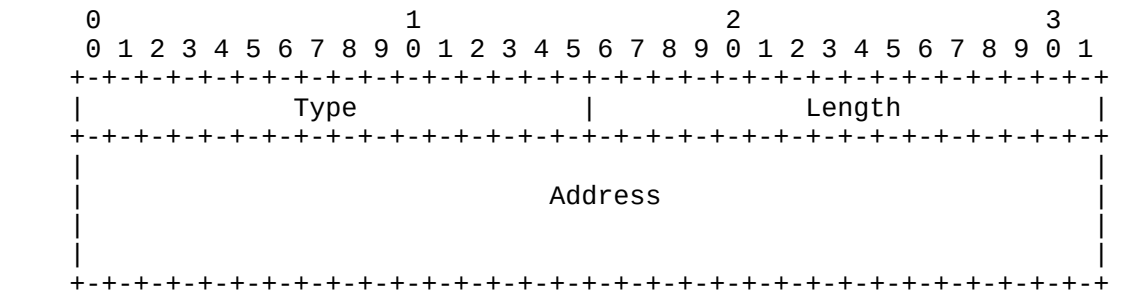
4.2.1. RVS_HMAC Parameter

The RVS_HMAC is a non-critical parameter whose only difference with the HMAC parameter defined in the HIP specification [RFC5201] is its "type" code. This change causes it to be located after the FROM parameter (as opposed to the HMAC):

Type	65500
Length	Variable. Length in octets, excluding Type, Length, and Padding.
HMAC	HMAC computed over the HIP packet, excluding the RVS_HMAC parameter and any following parameters. The HMAC is keyed with the appropriate HIP integrity key (HIP-lg or HIP-gl) established when rendezvous registration happened. The HIP "checksum" field MUST be set to zero, and the HIP header length in the HIP common header MUST be calculated not to cover any excluded parameter when the HMAC is calculated. The size of the HMAC is the natural size of the hash computation output depending on the used hash function.

To allow a rendezvous client and its RVS to verify the integrity of packets flowing between them, both SHOULD protect packets with an added RVS_HMAC parameter keyed with the HIP-lg or HIP-gl integrity key established while registration occurred. A valid RVS_HMAC SHOULD be present on every packet flowing between a client and a server and MUST be present when a FROM parameter is processed.

4.2.2. FROM Parameter

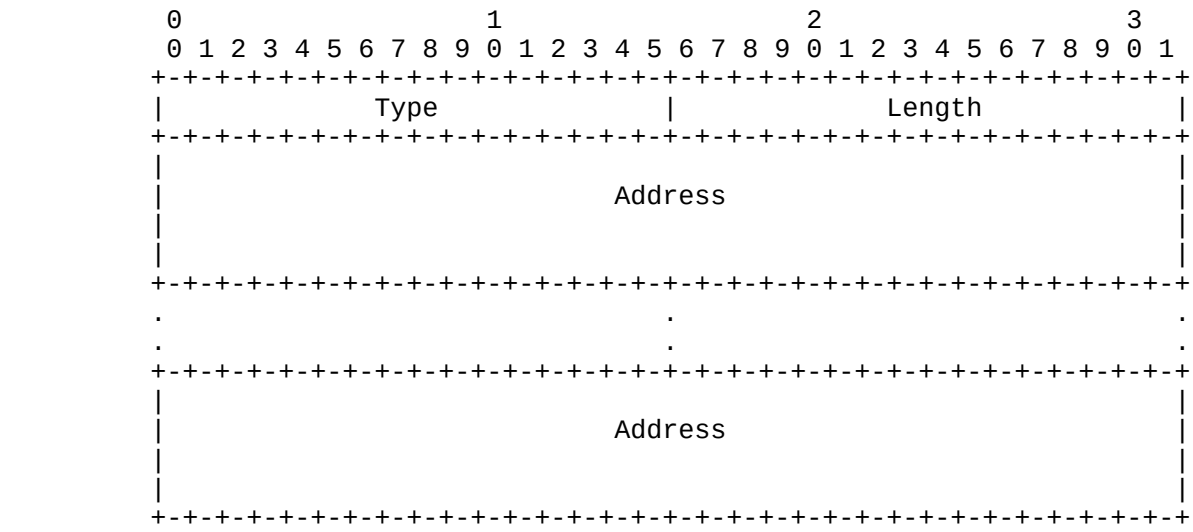


Type 65498
Length 16
Address An IPv6 address or an IPv4-in-IPv6 format IPv4 address.

A rendezvous server MUST add a FROM parameter containing the original source IP address of a HIP packet whenever the source IP address in the IP header is rewritten. If one or more FROM parameters are already present, the new FROM parameter MUST be appended after the existing ones.

Whenever an RVS inserts a FROM parameter, it MUST insert an RVS_HMAC protecting the packet integrity, especially the IP address included in the FROM parameter.

4.2.3. VIA_RVS Parameter



Type 65502
Length Variable
Address An IPv6 address or an IPv4-in-IPv6 format IPv4 address.

After the responder receives a relayed I1 packet, it can begin to send HIP packets addressed to the initiator's IP address, without further assistance from an RVS. For debugging purposes, it MAY include a subset of the IP addresses of its RVSs in some of these packets. When a responder does so, it MUST append a newly created VIA_RVS parameter at the end of the HIP packet. The main goal of using the VIA_RVS parameter is to allow operators to diagnose possible issues encountered while establishing a HIP association via an RVS.

4.3. Modified Packets Processing

The following subsections describe the differences of processing of I1 and R1 while a rendezvous server is involved in the base exchange.

4.3.1. Processing Outgoing I1 Packets

An initiator SHOULD NOT send an opportunistic I1 with a NULL destination HIT to an IP address that is known to be a rendezvous server address, unless it wants to establish a HIP association with the rendezvous server itself and does not know its HIT.

When an RVS rewrites the source IP address of an I1 packet due to egress filtering, it MUST add a FROM parameter to the I1 that contains the initiator's source IP address. This FROM parameter MUST be protected by an RVS_HMAC keyed with the integrity key established at rendezvous registration.

4.3.2. Processing Incoming I1 Packets

When a rendezvous server receives an I1 whose destination HIT is not its own, it consults its registration database to find a registration for the rendezvous service established by the HIT owner. If it finds an appropriate registration, it relays the packet to the registered IP address. If it does not find an appropriate registration, it drops the packet.

A rendezvous server SHOULD interpret any incoming opportunistic I1 (i.e., an I1 with a NULL destination HIT) as an I1 addressed to itself and SHOULD NOT attempt to relay it to one of its clients.

When a rendezvous client receives an I1, it MUST validate any present RVS_HMAC parameter. If the RVS_HMAC cannot be verified, the packet SHOULD be dropped. If the RVS_HMAC cannot be verified and a FROM parameter is present, the packet MUST be dropped.

A rendezvous client acting as responder SHOULD drop opportunistic I1s that include a FROM parameter, because this indicates that the I1 has been relayed.

4.3.3. Processing Outgoing R1 Packets

When a responder replies to an I1 relayed via an RVS, it MUST append to the regular R1 header a VIA_RVS parameter containing the IP addresses of the traversed RVSs.

4.3.4. Processing Incoming R1 Packets

The HIP specification [RFC5201] mandates that a system receiving an R1 MUST first check to see if it has sent an I1 to the originator of the R1 (i.e., the system is in state I1-SENT). When the R1 is replying to a relayed I1, this check SHOULD be based on HITs only. In case the IP addresses are also checked, then the source IP address MUST be checked against the IP address included in the VIA_RVS parameter.

5. Security Considerations

This section discusses the known threats introduced by these HIP extensions and the implications on the overall security of HIP. In particular, it argues that the extensions described in this document do not introduce additional threats to the Host Identity Protocol.

It is difficult to encompass the whole scope of threats introduced by rendezvous servers because their presence has implications both at the IP and HIP layers. In particular, these extensions might allow for redirection, amplification, and reflection attacks at the IP layer, as well as attacks on the HIP layer itself, for example, man-in-the-middle attacks against the HIP base exchange.

If an initiator has a priori knowledge of the responder's host identity when it first contacts the responder via an RVS, it has a means to verify the signatures in the HIP base exchange, which protects against man-in-the-middle attacks.

If an initiator does not have a priori knowledge of the responder's host identity (so-called "opportunistic initiators"), it is almost impossible to defend the HIP exchange against these attacks, because the public keys exchanged cannot be authenticated. The only approach would be to mitigate hijacking threats on HIP state by requiring an R1 answering an opportunistic I1 to come from the same IP address that originally sent the I1. This procedure retains a level of security that is equivalent to what exists in the Internet today.

However, for reasons of simplicity, this specification does not allow the establishment of a HIP association via a rendezvous server in an opportunistic manner.

6. IANA Considerations

This section is to be interpreted according to the Guidelines for Writing an IANA Considerations Section in RFCs [RFC2434].

This document updates the IANA Registry for HIP Parameters Types by assigning new HIP Parameter Types values for the new HIP Parameters defined in Section 4.2:

- RVS_HMAC (defined in Section 4.2.1)
- FROM (defined in Section 4.2.2)
- VIA_RVS (defined in Section 4.2.3)

This document defines an additional registration for the HIP Registration Extension [RFC5203] that allows registering with a rendezvous server for rendezvous service.

Number	Registration Type
1	RENDEZVOUS

7. Acknowledgments

The following people have provided thoughtful and helpful discussions and/or suggestions that have improved this document: Marcus Brunner, Tom Henderson, Miika Komu, Mika Kousa, Pekka Nikander, Justino Santos, Simon Schuetz, Tim Shepard, Kristian Slavov, Martin Stiemerling, and Juergen Quittek.

8. References

8.1. Normative References

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

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

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

[RFC5205] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", RFC 5205, April 2008.

8.2. Informative References

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3013] Killalea, T., "Recommended Internet Service Provider Security Services and Procedures", BCP 46, RFC 3013, November 2000.

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[RFC5206] Henderson, T., Ed., "End-Host Mobility and Multihoming with the Host Identity Protocol", RFC 5206, April 2008.

Authors' Addresses

Julien Laganier
DoCoMo Communications Laboratories Europe GmbH
Landsberger Strasse 312
Munich 80687
Germany

Phone: +49 89 56824 231
EMail: julien.ietf@laposte.net
URI: http://www.docomolab-euro.com/

Lars Eggert
Nokia Research Center
P.O. Box 407
Nokia Group 00045
Finland

Phone: +358 50 48 24461
EMail: lars.eggert@nokia.com
URI: http://research.nokia.com/people/lars_eggert/

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.