Internet Engineering Task Force (IETF)
Request for Comments: 8004
Obsoletes: 5204
Category: Standards Track
ISSN: 2070-1721

J. Laganier
Luminate Wireless, Inc.
L. Eggert
NetApp

October 2016

Host Identity Protocol (HIP) Rendezvous Extension

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 multihomed or mobile. This document obsoletes RFC 5204.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8004.

Table of Contents

1. Introduction

"The Host Identity Protocol (HIP) Architecture" [HIP-ARCH] 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 [RFC8003] 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 [RFC7401] and the HIP Registration Extension [RFC8003], this document defines and uses the following terms:

Rendezvous Service
: A HIP service provided by an RVS to its rendezvous clients. The RVS 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 an RVS.

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

3. Overview of Rendezvous Server Operation

Figure 1 shows a simple HIP base exchange without an RVS, 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 [RFC7401].

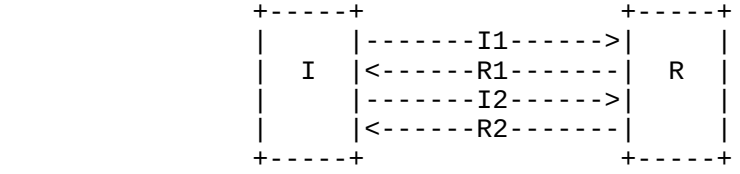


Figure 1: HIP Base Exchange without a Rendezvous Server

The End-Host Mobility and Multihoming with the HIP specification [HIP-HOST-MOB] 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 [HIP-ARCH] introduces RVSs 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 [RFC8005].

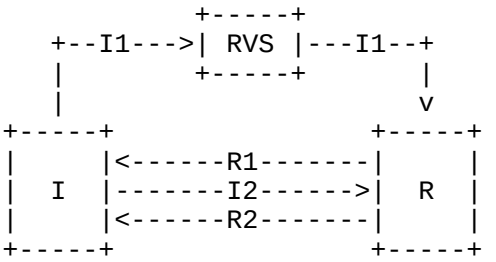


Figure 2: HIP Base Exchange with a Rendezvous Server

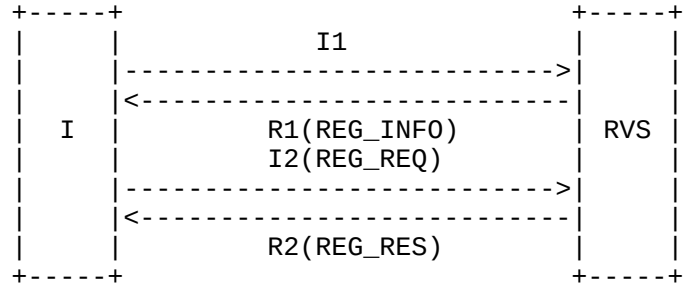
Figure 2 shows a HIP base exchange involving an RVS. It is assumed that HIP node R previously registered its HITs and current IP addresses with the RVS, using the HIP Registration Extension [RFC8003]. When the Initiator I tries to establish contact with the Responder R, it must send the I1 of the base exchange either to one of R's IP addresses (if known via DNS or other means) or to one of R's RVSs. Here, I obtains the IP address of R's RVS 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 (such as NAT and firewall traversal) to allow a client to initiate a base exchange through its own RVS. 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 Hashed Message Authentication Code (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 an RVS starts to relay HIP packets to a rendezvous client, the rendezvous client needs to register with the RVS to receive rendezvous service by using the HIP Registration Extension [RFC8003] as illustrated in the following schema:

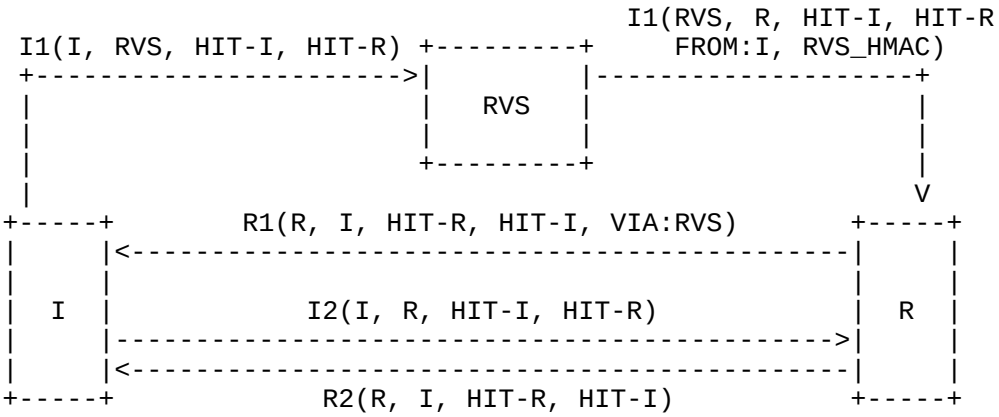


Rendezvous Client Registering with a Rendezvous Server

3.3. Relaying the Base Exchange

If a HIP node and one of its RVSs have a rendezvous registration, the RVSs 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 ingress filtering on the path from the RVS to the client [RFC2827] [RFC3013], a HIP RVS 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] [RFC6724]. 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 [RFC8003].



Rendezvous Server Rewriting IP Addresses

This modification of HIP packets at an RVS can be problematic because HIP 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 RVSs.

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 [RFC8003], allowing a HIP node to register with an RVS for rendezvous service and to notify the RVS aware of changes to its current location. It also describes an extension to the HIP specification [RFC7401] itself, allowing establishment of HIP associations via one or more HIP RVSs.

4.1. RENDEZVOUS Registration Type

This specification defines an additional registration for the HIP Registration Extension [RFC8003] that allows registering with an RVS 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 [RFC7401] is its "type" code. This change causes it to be located after the FROM parameter (as opposed to the HMAC):

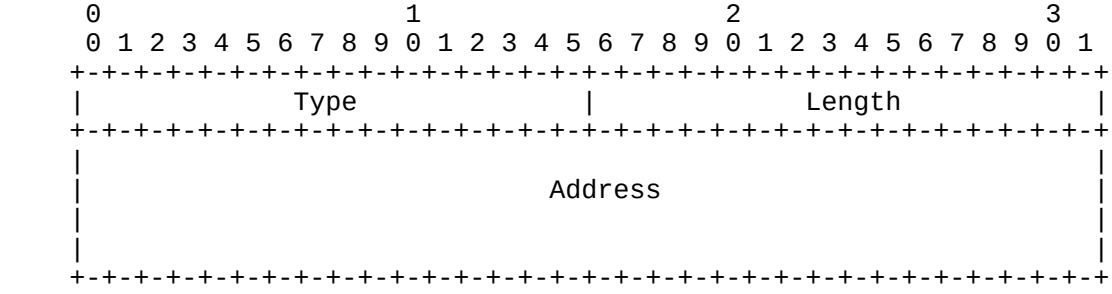
Type 65500

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

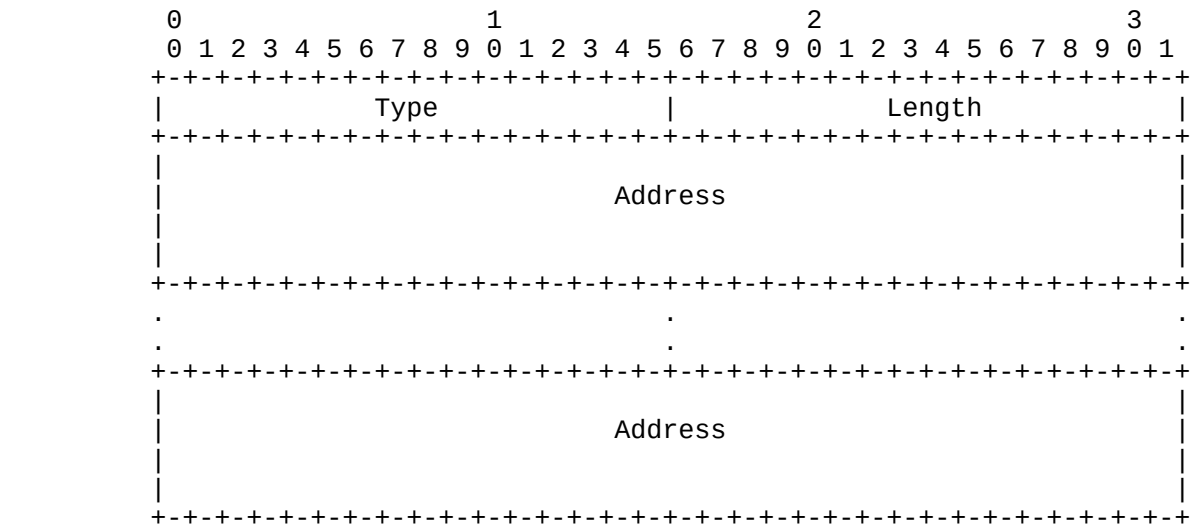


Type 65498
Length 16
Address An IPv6 address or an IPv4-in-IPv6 format IPv4 address.

An RVS 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 MUST append a newly created VIA_RVS parameter at the end of the R1 packet that contains the IP address of the RVS that relayed the I1 packet. Including more than one IP address in the VIA_RVS parameter is outside the scope of this specification. 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 the processing of I1 and R1 while an RVS 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 RVS 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 an RVS 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.

An RVS 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 [RFC7401] 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 HIP.

It is difficult to encompass the whole scope of threats introduced by RVSs 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 an RVS in an opportunistic manner.

6. IANA Considerations

[RFC5204], obsoleted by this document, made the following definitions and reservations in the "Parameter Types" subregistry under "Host Identity Protocol (HIP) Parameters":

Value	Parameter Type	Length
65498	FROM	16
65500	RVS_HMAC	variable
65502	VIA_RVS	variable

In the "Parameter Types" subregistry under "Host Identity Protocol (HIP) Parameters", references to [RFC5204] have been replaced by references to this document.

[RFC5204], obsoleted by this document, made the following definition and reservation in the "Registration Types" subregistry under "Host Identity Protocol (HIP) Parameters":

Value	Registration Type
1	RENDEZVOUS

In the "Registration Types" subregistry under "Host Identity Protocol (HIP) Parameters", references to [RFC5204] have been replaced by references to this document.

7. References

7.1. Normative References

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <http://www.rfc-editor.org/info/rfc6724>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <http://www.rfc-editor.org/info/rfc7401>.

[RFC8003] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", RFC 8003, DOI 10.17487/RFC8003, October 2016, <http://www.rfc-editor.org/info/rfc8003>.

[RFC8005] Laganier, J., "Host Identity Protocol (HIP) Domain Name System (DNS) Extension", RFC 8005, DOI 10.17487/RFC8005, October 2016, <http://www.rfc-editor.org/info/rfc8005>.

7.2. Informative References

[HIP-ARCH] Moskowitz, R. and M. Komu, "Host Identity Protocol Architecture", Work in Progress, draft-ietf-hip-rfc4423-bis-14, June 2016.

[HIP-HOST-MOB] Henderson, T., Vogt, C., and J. Arkko, "Host Mobility with the Host Identity Protocol", Work in Progress, draft-ietf-hip-rfc5206-bis-14, October 2016.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <http://www.rfc-editor.org/info/rfc2827>.

[RFC3013] Killalea, T., "Recommended Internet Service Provider Security Services and Procedures", BCP 46, RFC 3013, DOI 10.17487/RFC3013, November 2000, <http://www.rfc-editor.org/info/rfc3013>.

[RFC5204] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, DOI 10.17487/RFC5204, April 2008, <http://www.rfc-editor.org/info/rfc5204>.

Appendix A. Changes from RFC 5204

- Updated HIP references to revised HIP specifications.

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, Juergen Quittek, Justino Santos, Simon Schuetz, Tim Shepard, Kristian Slavov, and Martin Stiemerling.

Lars Eggert has received funding from the European Union's Horizon 2020 research and innovation program 2014-2018 under grant agreement No. 644866. This document reflects only the authors' views, and the European Commission is not responsible for any use that may be made of the information it contains.

Thanks to Joel M. Halpern for performing the Gen-ART review of this document as part of the publication process.

Authors' Addresses

Julien Laganier
Luminate Wireless, Inc.
Cupertino, CA
United States of America

Email: julien.ietf@gmail.com

Lars Eggert
NetApp
Sonnenallee 1
Kirchheim 85551
Germany

Phone: +49 151 12055791
Email: lars@netapp.com
URI: http://eggert.org