Internet Engineering Task Force (IETF)
Request for Comments: 7086
Category: Experimental
ISSN: 2070-1721

A. Keranen
G. Camarillo
J. Maenpaa
Ericsson

January 2014

Host Identity Protocol-Based Overlay Networking Environment (HIP BONE) Instance Specification for REsource LOcation And Discovery (RELOAD)

Abstract

This document is the HIP-Based Overlay Networking Environment (HIP BONE) instance specification for the REsource LOcation And Discovery (RELOAD) protocol. The document provides the details needed to build a RELOAD-based overlay that uses HIP.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. 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). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

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/rfc7086.

Table of Contents

1. Introduction

The HIP-Based Overlay Networking Environment (HIP BONE) specification [RFC6079] provides a high-level framework for building HIP-based [RFC5201] overlays. The HIP BONE framework does not address the specification of the details on how to combine a particular peer protocol with HIP to build an overlay. It leaves this up to documents referred to as HIP BONE instance specifications. As discussed in [RFC6079], a HIP BONE instance specification needs to define, minimally:

- the peer protocol to be used.
- what kind of Node IDs are used and how they are derived.
- which peer protocol primitives trigger HIP messages.
- how the overlay identifier is generated.

This document addresses all the previous items and provides additional details needed to build RELOAD-based HIP BONEs, referred to here as RELOAD HIP BONEs. The details on how different RELOAD modules would be integrated to a HIP implementation and what kind of APIs are used between them are left as implementation details or to be defined by other documents.

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]. In addition, this document uses the terms defined in [RFC5201], [RFC6079], [RFC6028], and [RFC6940].

3. Peer Protocol

The peer protocol to be used is REsource LOcation And Discovery (RELOAD) [RFC6940]. When used with RELOAD, HIP replaces the RELOAD's Forwarding and Link Management Layer (described in Section 6.5 of [RFC6940]).

4. Node ID Generation

This document specifies two modes for generating Node and Resource IDs. Which mode is used in an actual overlay is defined by the overlay configuration. Both of the modes are based on 16-byte ID mode of RELOAD; hence, only 16-byte RELOAD Node and Resource IDs MUST be used in a RELOAD HIP BONE.

HIP uses 128-bit Overlay Routable Cryptographic Hash Identifiers (ORCHIDs) [RFC4843] as identifiers. In a RELOAD HIP BONE, a peer's ORCHID can be used as a RELOAD Node ID (the "ORCHID" mode). In this mode, all the RELOAD IDs, including Resource IDs, are prefixed with the ORCHID prefix, and the lower 100 bits of the IDs defined by RELOAD usage documents are used after the prefix.

In the other Node ID mode, namely "RELOAD", all 128 bits are generated as defined in [RFC6940]. This results in a larger usable address space than using the ORCHID mode, but the resulting Node IDs cannot be used with legacy applications and APIs, as discussed in Section 5.1 of [RFC6079].

5. Mapping between Protocol Primitives and HIP Messages

RELOAD HIP BONE replaces the RELOAD protocol primitives taking care of connection establishment with the HIP base exchange, whereas the rest of the RELOAD messages are conveyed within HIP messages. The Forwarding and Link Management Layer functionality of RELOAD, including all the NAT traversal functionality, is replaced by HIP, existing extensions of HIP, and the extensions defined in this document.

The standard RELOAD messages consist of three parts: the forwarding header, the message contents, and the security block. When RELOAD messages are sent in a RELOAD HIP BONE overlay, the RELOAD message contents are used as such within HIP DATA [RFC6078] messages, but the functionality of the forwarding header and security block are replaced with the HIP header, HIP Destination and Via lists [RFC6028], CERT [RFC6253], TRANSACTION_ID [RFC6078], and the OVERLAY_ID and OVERLAY_TTL [RFC6079] parameters, as defined in the following sections.

5.1. Forwarding Header

The RELOAD forwarding header is used for forwarding messages between peers and to their final destination. The forwarding header’s overlay field value MUST be used as such in an OVERLAY_ID parameter and the transaction_id field in a TRANSACTION_ID parameter. That is, all RELOAD HIP BONE messages MUST contain these parameters; and, the length of the OVERLAY_ID parameter’s identifier field is 4, and the length of the TRANSACTION_ID parameter is 8 octets. HIP Destination and Via lists are used for the same purpose as the destination_list and via_list in the forwarding header, with the exception that all Resource IDs MUST be of the same length as Node IDs, and compressed IDs MUST NOT be used. The Time to Live (TTL) value in the OVERLAY_TTL parameter is used like the ttl field in the forwarding header.

The functionality of the fragment and length fields are provided by the HIP headers. The relo_token, version, and max_response_length are not needed with HIP. The forwarding header’s options field, if needed eventually for some extensions, can be substituted with additional HIP parameters.

5.2. Security Block

The RELOAD security block contains certificates and digital signatures of the message. All the HIP DATA messages are digitally signed by the originator of the message and contain the HOST_ID parameter with the identifier that can be used for verifying the signature. Certificates are delivered in a HIP CERT parameter as defined in [RFC6253] or stored to the overlay using the RELOAD Certificate Storage Usage.

Note that when the RELOAD mode for Node ID generation is used, the certificate certifying that a host is allowed to use a certain Node ID MUST contain the host's Node ID instead of the Host Identity Tag (HIT) in the "subjectAltName" field of the certificate as described in Section 11.3 of [RFC6940], while the "Subject" field contains the HIT calculated from the Host Identity.

5.3. Replaced RELOAD Messages

The Attach procedure in RELOAD establishes a connection between two peers. This procedure is performed using the AttachReq and AttachAns messages. When HIP is used, the Attach procedure is performed by using a HIP base exchange. That is, peers send HIP first Initiator (I1) messages instead of RELOAD AttachReq messages. This behavior replaces the one described in Section 6.5 of [RFC6940].

The AppAttach procedure in RELOAD is used for creating a connection for other applications than RELOAD. Also, the AppAttach procedure is replaced with the HIP base exchange, and after the base exchange, peers can exchange any application layer data using the normal transport layer ports over the NAT traversing IPsec connection.

This specification does not support flooding of configuration files, so ConfigUpdate requests and responses (Section 6.5.4 of [RFC6940]) MUST NOT be sent in the overlay. RELOAD Ping messages (Section 6.5.3 of [RFC6940]) MAY be used.

For all other RELOAD messages, the message contents are used as such within HIP DATA messages.

6. Securing Communication

RELOAD uses Transport Layer Security (TLS) [RFC5246] connections for securing the hop-by-hop messaging and certificates and signatures for providing integrity protection for the overlay messages and for the data stored in the overlay.

With a RELOAD HIP BONE, instead of using TLS connections as defined in [RFC6940], all HIP overlay messages MUST be sent using encrypted connections [RFC6261].

The data objects stored in the RELOAD HIP BONE overlay are signed, and the signatures are stored as defined in [RFC6940] with the exception that SignerIdentity is carried in the HIP DATA message's HOST_ID parameter instead of using the RELOAD security block. Where certificates are needed, they are sent using the HIP CERT parameter.

7. Routing HIP Messages via the Overlay

If a host has no valid locator for the receiver of a new HIP packet, and the receiver is part of a RELOAD HIP BONE overlay the host is participating in, the host can send the HIP packet to the receiver using the overlay routing.

When sending a HIP packet via the overlay, the host MUST add an empty ROUTE_VIA parameter [RFC6028] to the packet with the SYMMETRIC and MUST_FOLLOW flags set and an OVERLAY_ID parameter containing the identifier of the right overlay network. The host consults the RELOAD Topology Plugin for the next hop and sends the HIP packet to that host.

An intermediate host receiving a HIP packet with the OVERLAY_ID parameter checks if it is participating in that overlay and SHOULD drop packets sent to unknown overlays. If the host is not the final destination of the packet (i.e., the Receiver HIT in the HIP header does not match to any of its HITs), it checks if the packet contains a ROUTE_DST parameter. Such packets are forwarded to the next hop as specified in [RFC6028]. If the packet does not contain a ROUTE_DST parameter, the host finds the next hop from the RELOAD Topology Plugin and forwards the packet there. As specified in [RFC6028], the host adds the HIT it uses on the HIP association with the next-hop host to the end of the ROUTE_VIA parameter, if present.

When the final destination host receives the HIP packet, the host processes it as specified in [RFC5201]; and, if the packet is a HIP DATA packet, the contents are processed as specified in [RFC6940]. If the HIP packet generates a response, the response is routed back on the same path using the ROUTE_DST parameter as specified in [RFC6028].

8. Enrollment and Bootstrapping

The RELOAD HIP BONE instance uses the enrollment and bootstrap procedure defined by RELOAD [RFC6940] with the exceptions listed below.

- In RELOAD, a node wishing to enroll in an overlay starts with obtaining a configuration document as explained in [RFC6940]. This specification extends the RELOAD overlay configuration document as defined in Section 10.

- The X.509 certificates used by the RELOAD HIP BONE instance are similar to those of RELOAD except that they contain HITs instead of RELOAD URIs. The HITs are included in the SubjectAltName field of the certificate as described in [RFC6253].

- When contacting a bootstrap node, instead of forming a Datagram Transport Layer Security (DTLS) or TLS connection, the host MUST perform a HIP base exchange with the bootstrap node. The base exchange MAY be performed using a HIP rendezvous or relay server.

9. NAT Traversal

RELOAD relies on the Forwarding and Link Management Layer providing NAT traversal capabilities. Thus, the RELOAD HIP BONE instance implementations MUST implement some reliable NAT traversal mechanism. To maximize interoperability, all implementations SHOULD implement at least [RFC5770].

HIP relay servers are not necessarily needed with this HIP BONE instance since the overlay network can be used for relaying the base exchange, and further HIP signaling can be done directly between the peers. However, if it is possible that a bootstrap peer is behind a NAT, it MUST register with a HIP relay so that there is a reliable way to connect to it.

10. RELOAD Overlay Configuration Document Extension

This document modifies the bootstrap-node element of the RELOAD overlay configuration document. The modified bootstrap-node element contains the following attributes:

address: The locator of the bootstrap node.

port: The HIP port of the bootstrap node.

hit: The HIT of the bootstrap node.

If the bootstrap-node element does not contain a HIT, the opportunistic mode (as specified in [RFC5201]) SHOULD be used for contacting the bootstrap node. If the element does not contain a port number, the bootstrap node SHOULD be contacted by starting the base exchange as defined in [RFC5201]. Otherwise, the base exchange MUST be started with UDP encapsulation, as defined in [RFC5770], using the given port as the destination port number.

A RELOAD HIP BONE overlay MUST use the Overlay Link Protocol type "HIP" in the configuration document's overlay-link-protocol element. The enrolling node MUST check the overlay-link-protocol element and proceed with procedures defined in this document only if the "HIP" link type is found.

This document also adds a new element inside the configuration element that defines which mode (see Section 4) is used for generating the Node and Resource IDs. The name of the element is "hipbone-id-mode" and the content is the identifier of the mode: "ORCHID" for the ORCHID prefixed IDs and "RELOAD" for the IDs that use the whole 128 bits as defined by the RELOAD specification. The NodeIdLength MUST be set to 16 in the configuration document, and the 16 bytes are used, depending on the generation mode, as defined in Section 4.

11. Security Considerations

The security considerations of RELOAD (Section 13 of [RFC6940]), with the exception of TLS-specific features, also apply to RELOAD HIP BONE instances.

Limiting the Node ID and Resource ID space into 128 bits (or 100 bits with ORCHID prefixes) results in a higher probability for ID collisions, both unintentional and intentional, than using larger address spaces.

12. IANA Considerations

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

IANA has updated the "RELOAD Overlay Link Protocol Registry" [RFC6940] by assigning the new Overlay Link Protocol type "HIP" (see Section 10).

13. Acknowledgements

Tom Henderson, Spencer Dawkins, and Christer Holmberg have provided valuable reviews and comments on the document.

14. References

14.1. Normative References

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

[RFC4843] Nikander, P., Laganier, J., and F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID)", RFC 4843, April 2007.

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

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

[RFC5770] Komu, M., Henderson, T., Tschofenig, H., Melen, J., and A. Keranen, "Basic Host Identity Protocol (HIP) Extensions for Traversal of Network Address Translators", RFC 5770, April 2010.

[RFC6028] Camarillo, G. and A. Keranen, "Host Identity Protocol (HIP) Multi-Hop Routing Extension", RFC 6028, October 2010.

[RFC6078] Camarillo, G. and J. Melen, "Host Identity Protocol (HIP) Immediate Carriage and Conveyance of Upper-Layer Protocol Signaling (HICCUPS)", RFC 6078, January 2011.

[RFC6079] Camarillo, G., Nikander, P., Hautakorpi, J., Keranen, A., and A. Johnston, "HIP BONE: Host Identity Protocol (HIP) Based Overlay Networking Environment (BONE)", RFC 6079, January 2011.

[RFC6253] Heer, T. and S. Varjonen, "Host Identity Protocol Certificates", RFC 6253, May 2011.

[RFC6261] Keranen, A., "Encrypted Signaling Transport Modes for the Host Identity Protocol", RFC 6261, May 2011.

[RFC6940] Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", January 2014.

14.2. Informative References

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Authors' Addresses

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

EMail: Ari.Keranen@ericsson.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Jouni Maenpaa
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Jouni.Maenpaa@ericsson.com