HIP Working Group
Internet-Draft
Intended status: Experimental
Expires: April 21, 2011

A. Keranen
Ericsson
October 18, 2010

Host Identity Protocol Signaling Message Transport Modes
draft-ietf-hip-over-hip-02

Abstract

This document specifies two transport modes for Host Identity Protocol (HIP) signaling messages that allow conveying them over encrypted connections initiated with the Host Identity Protocol.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Host Identity Protocol (HIP) [RFC5201] signaling messages can be exchanged over plain IP using the protocol number reserved for this purpose, or over UDP using the UDP port reserved for HIP NAT traversal [RFC5770]. When two hosts perform a HIP base exchange, they set up an encrypted connection between them for data traffic, but continue to use plain IP or UDP for HIP signaling messages.

This document defines how the encrypted connection can be used also for HIP signaling messages. Two different modes are defined: HIP over Encapsulating Security Payload (ESP) and HIP over TCP. The benefit of sending HIP messages over ESP is that all signaling traffic (including HIP headers) will be encrypted. If HIP messages are sent over TCP (which in turn is transported over ESP), TCP can handle also message fragmentation where needed.

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].

3. Protocol Extensions

This section defines how support for different HIP signaling message transport modes is negotiated and the normative behavior required by the extension.

3.1. Mode Negotiation in HIP Base Exchange

A HIP host implementing this specification SHOULD indicate the modes it supports, and is willing to use, in the base exchange. The HIP signaling message transport mode negotiation is similar to HIP NAT traversal mode negotiation: first the Responder lists the supported modes in a HIP_TRANSPORT_MODE parameter (see Figure 1) in the R1 packet. The modes are listed in priority order; the more preferred mode(s) first. If the Initiator supports, and is willing to use, any of the modes proposed by the Responder, it selects one of the modes by adding a HIP_TRANSPORT_MODE parameter containing the selected mode to the I2 packet. Finally, if the Initiator selected one of the modes and the base exchange succeeds, hosts MUST use the selected mode for the following HIP signaling messages sent between them for the duration of the HIP association or until another mode is negotiated.

If the Initiator cannot or will not use any of the modes proposed by the Responder, the Initiator SHOULD include an empty HIP_TRANSPORT_MODE parameter to the I2 packet to signal that it support this extension but will not use any of the proposed modes. Depending on local policy, the Responder MAY either abort the base exchange or continue HIP signaling without using an encrypted connection, if there was no HIP_TRANSPORT_MODE parameter in I2 or the parameter was empty. If the Initiator selects a mode that the Responder does not support (and hence was not included in R1), the Responder MUST abort the base exchange. If the base exchange is aborted due to (possibly lack of) HIP_TRANSPORT_PARAMETER, the Responder SHOULD send a NO_VALID_HIP_TRANSPORT_MODE NOTIFY packet (see Section 4) to the Initiator.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             Type              |             Length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           Mode ID #1          |            Mode ID #2         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           Mode ID #n          |             Padding           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type   [TBD by IANA; 7680]
Length   length in octets, excluding Type, Length, and Padding
Mode ID defines the proposed or selected transport mode(s)

The following Mode IDs are defined:

ID name	Value
RESERVED	0
DEFAULT	1
ESP	2
ESP-TCP	3

Figure 1: Format of the HIP_TRANSPORT_MODE parameter

The mode DEFAULT indicates that the same transport mode (e.g., plain IP or UDP) that was used for the base exchange should be used for subsequent HIP signaling messages. In the ESP mode the messages are sent as such on the encrypted ESP connection and in the ESP-TCP mode TCP is used within the ESP tunnel.

3.2. Mode Negotiation After HIP Base Exchange

If a HIP hosts wants to change to a different transport mode (or start using a transport mode) some time after the base exchange, it sends a HIP UPDATE packet with a HIP_TRANSPORT_MODE parameter containing the mode(s) it would prefer to use. The host receiving the UPDATE SHOULD respond with an UPDATE packet containing the mode that is selected as in the negotiation during the base exchange. If the receiving host does not support, or is not willing to use, any of the listed modes, it SHOULD respond with an UPDATE packet where the HIP_TRANSPORT_MODE parameter contains only the currently used transport mode (even if one was not included in the previous UPDATE packet) and continue using that mode.

Since the HIP_TRANSPORT_MODE parameter's type is not critical (as defined in Section 5.2.1 of [RFC5201]), a host not supporting this extension would simply reply with an acknowledgement UPDATE packet without a HIP_TRANSPORT_MODE parameter. In such a case, depending on local policy as in mode negotiation during the base exchange, the host that requested the new transport mode MAY close the HIP association. If the association is closed, the host closing the association SHOULD send a NO_VALID_HIP_TRANSPORT_MODE NOTIFY packet to the other host before closing the association.

3.3. HIP Messages on Encrypted Connections

This specification defines two different transport modes for sending HIP packets over encrypted ESP connections. These modes require that the ESP transport format [RFC5202] is negotiated to be used between the hosts. If the ESP transport format is not used, these modes MUST NOT be offered in the HIP_TRANSPORT_MODE parameter. If a HIP_TRANSPORT_MODE parameter containing an ESP transport mode is received but the ESP transport format is not used, a host MUST NOT select such a mode but act as specified in Section 3.1 (if performing a base exchange) or Section 3.2 (if performing an UPDATE) when no valid mode is offered.

The ESP mode provides simple protection for all the signaling traffic and can be used as a generic replacement for the DEFAULT mode in cases where all signaling traffic should be encrypted. If the HIP messages may become so large that they would need to be fragmented, e.g., because of HIP certificates [I-D.ietf-hip-cert] or DATA messages [I-D.ietf-hip-hiccups], it is RECOMMENDED to use the ESP-TCP mode which can handle message fragmentation at TCP level instead of relying on IP level fragmentation.

HIP messages that result in changing or generating new keying material, i.e., the base exchange and re-keying UPDATE messages, MUST

NOT be sent over an encrypted connection that is created using the keying material that is being changed, nor over an encrypted connection using the newly created keying material.

3.3.1. ESP mode

If the ESP mode is selected in the base exchange, both hosts MUST listen for incoming HIP signaling messages and send outgoing messages on the encrypted connection. The ESP header's next header value for such messages MUST be set to HIP (139).

3.3.2. ESP-TCP mode

If the ESP-TCP mode is selected, the host with the larger HIT (calculated as defined in Section 6.5 of [RFC5201]) MUST start to listen for an incoming TCP connection on the port 10500 on the encrypted connection and the other host MUST create a TCP connection to that port. The host with the smaller HIT SHOULD use port 10500 as the source port for the TCP connection. Once the TCP connection is established, both hosts MUST listen for incoming HIP signaling messages and send the outgoing messages using the TCP connection. The ESP next header value for messages sent using the ESP-TCP mode connections MUST be set to TCP (6).

If the hosts are unable to create the TCP connection, the host that initiated the mode negotiation MUST restart the negotiation with UPDATE message and SHOULD NOT propose the ESP-TCP mode. If local policy does not allow using any other mode than ESP-TCP, the HIP association MUST be closed. The UPDATE or CLOSE message MUST be sent using the same transport mode that was used for negotiating the use of the ESP-TCP mode.

Since TCP provides reliable transport, the HIP messages sent over TCP MUST NOT be retransmitted for the purpose of achieving reliable transmission. Instead, a host SHOULD wait to detect that the TCP connection has failed to retransmit the packet successfully in a timely manner (such detection is platform- and policy-specific) before concluding that there is no response.

3.4. Recovering from Failed Encrypted Connections

If the encrypted connection fails for some reason, it can no longer be used for HIP signaling and the hosts SHOULD re-establish the connection using HIP messages that are sent outside of the encrypted connection. Hence, while listening for incoming HIP messages on the encrypted connection, hosts MUST still accept incoming HIP messages using the same transport method (e.g., UDP or plain IP) that was used for the base exchange. When responding to a HIP message sent outside

of encrypted connection, the response MUST be sent using the same transport method as the original message used. Hosts SHOULD send outside of the encrypted connection only HIP messages that are used to reestablish the encrypted connection. Especially, messages that are intended to be sent only encrypted (e.g., DATA messages using an encrypted transport mode) MUST NOT be sent before the encrypted connection is reestablished.

The UPDATE messages used for re-establishing the encrypted connection MUST contain a HIP_TRANSPORT_MODE parameter and the negotiation proceeds as described in Section 3.2.

3.5. Host Mobility

If the host's address changes, it may not be able to send the mobility UPDATE messages using the encrypted connection before it breaks. This results in a similar situation as if the encrypted connection had failed and the hosts need to re-negotiate the new addresses using un-encrypted UPDATE messages and possibly rendezvous [RFC5204] or HIP relay [RFC5770] servers. Also these UPDATE messages MUST contain the HIP_TRANSPORT_MODE parameter and perform the transport mode negotiation.

4. Notify Packet Types

The new Notify Packet Type [RFC5201] defined in this document is shown below. The Notification Data field for the error notifications SHOULD contain the HIP header of the rejected packet.

NOTIFICATION PARAMETER - ERROR TYPES	Value
NO_VALID_HIP_TRANSPORT_MODE	[TBD by IANA;100]

If a host sends an UPDATE message that does not have any transport mode the receiving host is willing to use, the receiving host sends back a NOTIFY error packet with this type.

5. Security Considerations

By exchanging the HIP messages over ESP connection, all HIP signaling data (after the base exchange but excluding keying material (re)negotiation) will be encrypted, but only if NULL encryption is not used. Thus, a host requiring confidentiality for the HIP signaling messages must check that encryption is negotiated to be used on the ESP connection. Moreover, the level of protection

provided by the ESP transport modes depends on the selected ESP transform; see [RFC5202] and [RFC4303] for security considerations of the different ESP transforms.

6. Acknowledgements

Thanks to Gonzalo Camarillo, Kristian Slavov, Tom Henderson, Miika Komu, and Jan Melen for comments on the draft.

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 HIP_TRANSPORT_MODE parameter (defined in Section 3.1).

The HIP_TRANSPORT_MODE parameter has 16-bit unsigned integer fields for different modes, for which IANA is to create and maintain a new sub-registry entitled "HIP Transport Modes" under the "Host Identity Protocol (HIP) Parameters" registry. Initial values for the transport mode registry are given in Section 3.1; future assignments are to be made through IETF Review [RFC5226]. Assignments consist of a transport mode identifier name and its associated value.

This document also defines new HIP Notify Packet Type [RFC5201] NO_VALID_HIP_TRANSPORT_MODE in Section 4.

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.

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

8.2. Informational References

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC5204] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, April 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.

[I-D.ietf-hip-cert]
Heer, T. and S. Varjonen, "HIP Certificates", draft-ietf-hip-cert-04 (work in progress), September 2010.

[I-D.ietf-hip-hiccups]
Camarillo, G. and J. Melen, "HIP (Host Identity Protocol) Immediate Carriage and Conveyance of Upper- layer Protocol Signaling (HICCUPS)", draft-ietf-hip-hiccups-05 (work in progress), July 2010.

Author's Address

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

Email: Ari.Keranen@ericsson.com