HIP
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2017

R. Moskowitz
X. Xu
B. Liu
Huawei

June 26, 2017

Encapsulation of IP within IP managed by HIP
draft-moskowitz-hip-ipnhip-02.txt

Abstract

This document defines how to encapsulate IP within IP when the tunnel is managed with HIPv2 [RFC7401]. The goal is reduced header size and improved security over IPnIP [RFC2003] and [RFC2004].

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on December 28, 2017.

Table of Contents

1. Introduction

MobileIP has opted for a simple IP within IP tunneling mechanism without any tunnel security. The justification for this approach over secure tunneling mechanisms like ESP [RFC4303] is outside the scope of this document. The approach here is to define a IPnIP header that leverages the HIP Security Association and is potentiality smaller than RFC2004 [RFC2004] as well as provides for a higher security posture.

The IPnHIP header defined here also supports the per-packet compression, GPCOMP [I-D.moskowitz-gpcomp], which offers further gains in transmission efficiency.

Implementors are expected to be familiar with both HIPv2 and ESP with HIP [RFC7402]. This document draws heavily on RFC7402 to the extent that much of the flow process is not duplicated here.

2. Terms and Definitions

2.1. Requirements 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].

2.2. Definitions

IPnHIP: The short name for IP-within-IP managed by HIP.

SPI: The Security Parameter Index.

3. The advantage of a HIP managed IP-within-IP Tunnel

HIP maps the peer address pair into two 32 bit uni-directional Security Parameter Indexes (SPI). It is only necessary for a tunnel to include the SPI that indicates the traffic direction. The HIP layer provides the translation between the SPI and the addresses. The resultant header is thus almost always smaller than with RFC2004.

This results that an attacker will have to learn about this SPI to addressing mapping to execute an attack against the higher layers within the tunnel.

The addition of an ESP-styled sequence number further reduces the attack window as the attacker must know the current sequence number window. The inclusion of a 32 bit sequence number enlarges the header, but for IPv4 it is still in line with the size for RFC2004 and for IPv6 it is still considerably smaller.

4. IPnHIP Header Format

The Protocol field in the IP header is replaced by protocol number TBD for the IPnHIP encapsulation protocol.

The format of the IP-within-IP-with-HIP header is as follows:

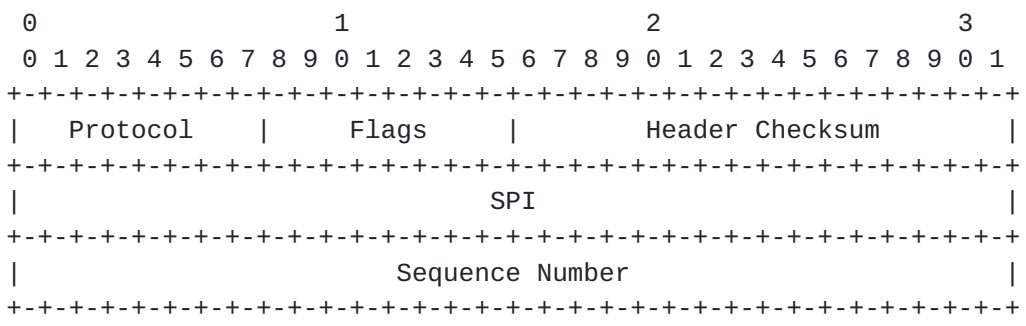


Figure 1 - Header Format

Protocol

Copied from the Protocol field in the original IP header.

Flags

Is a set of 8 options flags.

Header Checksum

The 16-bit one's complement of the one's complement sum of all 16-bit words in this header. For purposes of computing the checksum, the value of the checksum field is 0. The IP header and IP payload (after the minimal forwarding header) are not included in this checksum computation.

SPI

The SPI as defined in section SPI.

Sequence Number

As defined in RFC 4303.

Flags is a set of 8 options flags. Bit 7 is the GPComp [I-D.moskowitz-gpcomp] bit compression option bit.

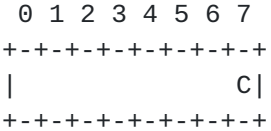


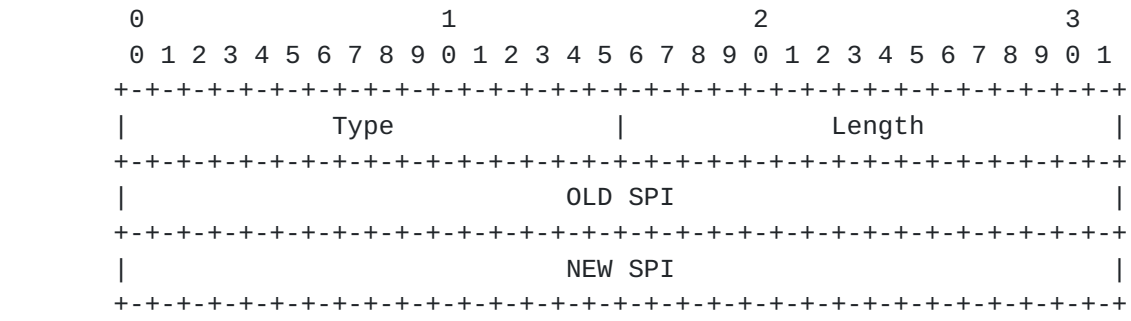
Figure 2 - Flags Field

5. HIP parameters to negotiate IPnHIP

Two HIP parameters are defined for setting up IPnHIP tunnel format associations in HIP communication and for restarting existing ones.

Parameter	Type	Length	Data
IPnHIP_INFO	[TBD-IANA]	8	Remote's old SPI, new SPI
IPnHIP_TRANSFORM	[TBD-IANA]	variable	IP Encapsulation in IP

5.1. IPnHIP_INFO

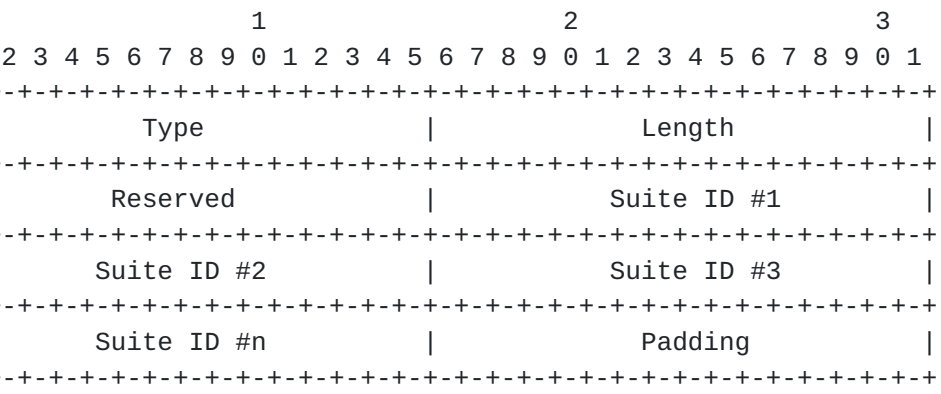


Type	[TBD-IANA]
Length	8
OLD SPI	old SPI for data sent to address(es) associated with this SA. If this is an initial SA setup, the OLD SPI value is zero.
NEW SPI	new SPI for data sent to address(es) associated with this SA.

The processing of IPnHIP_INFO is similar to ESP_INFO, section 5.1.1 of RFC7402 [RFC7402], without the KEYMAT generation.

5.2. IPnHIP_TRANSFORM

The IPnHIP_TRANSFORM parameter is used during IPnHIP SA establishment. The first party sends a selection of transform families in the IPnHIP_TRANSFORM parameter, and the peer must select one of the proposed values and include it in the response IPnHIP_TRANSFORM parameter.



Type	[TBD-IANA]
Length	length in octets, excluding Type, Length, and padding.
Reserved	zero when sent, ignored when received.
Suite ID	defines the IPnHIP Suite to be used.

The following Suite IDs can be used:

Suite ID	Value	
RESERVED	0	[this draft]
IPnHIP	1	[sec IPnHIP1]

The sender of an IPnHIP transform parameter MUST make sure that there are no more than six (6) Suite IDs in one IPnHIP transform parameter. Conversely, a recipient MUST be prepared to handle received transform parameters that contain more than six Suite IDs. The limited number of Suite IDs sets the maximum size of the IPnHIP_TRANSFORM parameter. As the default configuration, the IPnHIP_TRANSFORM parameter MUST contain at least one of the mandatory Suite IDs. There MAY be a configuration option that allows the administrator to override this default.

Currently only one IPnHIP_TRANSFORM is defined. Future work may define others.

6. HIP IPnHIP Security Association Setup

The IPnHIP Security Association is set up during the base exchange. The following subsections define the IPnHIP SA setup procedure using both base exchange messages (R1, I2, R2) and UPDATE messages.

The IPnHIP Security Association follows the same process as that of the ESP Security Association (sec 5.2 RFC7402 [RFC7402] except for the KEYMAT.

IPnHIP SA does not have any keying material, and thus those processes are not needed. 'Rekeying' to assign new SPIs is still needed to manage the sequence numbering.

7. ICMP Messages

ICMP Message handling is the same as sec 5.4 RFC7402 [RFC7402].

8. Packet Processing

Packet processing is mainly defined in sec 6 RFC7402 [RFC7402] with following changes.

8.1. Packet Compression

Packet compression is negotiated by HIP using the GPCOMP_INFO parameter defined in [I-D.moskowitz-ssls-hip].

IPnHIP uses the Implied Structure of GPCOMP [I-D.moskowitz-gpcomp] and follows the Compress/Uncompresing process defined there in.

8.2. Processing Application Data

IPnHIP sequence number processing follows RFC4303 [RFC4303]. With the extended 64 bit sequence number, the rarely will be the need to update the SPI to reset the sequence number. Any resetting the SPI will be driven by privacy concerns. The rest of the packet processing follows RFC2004 [RFC2004].

8.3. Processing HIP Packets

HIP packet processing is the same as sec 6 RFC7402 [RFC7402] without the keying parameter handling.

9. ESP or Minimal IPnIP or IPnHIP

There are at least three ways to compare these encapsulation protocols:

- Encapsulation cost in bytes
- Encapsulation cost in processing
- Security posture

9.1. Encapsulation cost in bytes

From the analysis below, IPnHIP is consistently the cheapest option.

- ESP adds 9 bytes + pad (0 - 3 bytes) + ICV. For GMAC-96 this is 17 bytes + pad.
- Minimal IPnIP is 4 bytes + 2 * IP address length. For IPv4 this is 12 bytes and for IPv6 36 bytes.
- IPnHIP is 12 bytes (Note: Can use GPCOMP with Implied Structure, i.e. no header cost, for further savings.)

9.2. Encapsulation cost in processing

- ESP with GMAC-96 is perhaps the computationally lightest transform. GMAC has only 2 AES operations + n GHASH operations.
- Minimal IPnIP has no cryptographic processing overhead.
- IPnHIP has no cryptographic processing overhead. GPCOMP does add the compression processing.

9.3. Security posture

- ESP traffic is fully protected up to the strength of the cryptographic transform used. Plus the HIP SA is protection against MITM attacks provided there is authentication of the HITs used.
- Minimal IPnIP has no security protection. A party that discovers IPnIP flow can interject any traffic desired.
- IPnHIP masks the internal identities by only including the HIP SA SPIs and a sequence number. This presents a number of challenges to an attacker. They have to know the sequence number window and what the SPI maps to. This is not as strong as ESP, but more protection that IPnIP provides.

10. IANA Considerations

The IP protocol number of NN for IPnHIP is assigned by IANA.

The following change to the "Host Identity Protocol (HIP) Parameters" registries has been made:

IPnHIP_INFO: This document defines the new IPnHIP_INFO parameter (see Section 5.1). The parameter value will be assigned by IANA. Its value should come from the 66-127 range.

IPnHIP_TRANSFORM: This document defines the new IPnHIP_TRANSFORM parameter (see Section 5.2). The parameter value will be assigned by IANA. Its value should come from the 4096-4480 range.

11. Security Considerations

IPnHIP lacks the protections provided by ESP. ESP with the GMAC transform should be seriously considered for a fast, Integrity only mode instead of IPnHIP. GMAC has only 2 AES block operations per ESP payload.

There are policy cases where only a non-securable tunnel will be permitted. IPnHIP provides a high level of tunnel management security through HIP and better privacy and spoofing and replay resiliency than IPnIP due to its use of a sequence number scheme and an SPI instead of the internal IP addresses.

HIP fast Mobility provides the high trust provided by HIP for address remapping without needing triangular data routing.

GPCOMP, like ESP, is not believed to be subject to the TLS BEAST attack.

12. Acknowledgments

Sue Hares of Huawei contributed to the clarity in this document.

13. References

13.1. Normative References

[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>.

[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>.

13.2. Informative References

[I-D.moskowitz-gpcomp]
Moskowitz, R., Hares, S., Faynberg, I., Lu, H., and P. Giacomin, "GPCOMP", draft-moskowitz-gpcomp-01 (work in progress), October 2016.

[I-D.moskowitz-ssls-hip]
Moskowitz, R., Xia, L., Faynberg, I., Hares, S., and P. Giacomin, "Secure Session Layer Services KMP via HIP", draft-moskowitz-ssls-hip-01 (work in progress), October 2016.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <http://www.rfc-editor.org/info/rfc2003>.

[RFC2004] Perkins, C., "Minimal Encapsulation within IP", RFC 2004, DOI 10.17487/RFC2004, October 1996, <http://www.rfc-editor.org/info/rfc2004>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC7402] Jokela, P., Moskowitz, R., and J. Melen, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)", RFC 7402, DOI 10.17487/RFC7402, April 2015, <http://www.rfc-editor.org/info/rfc7402>.

Authors' Addresses

Robert Moskowitz
Huawei
Oak Park, MI 48237
USA

Email: rgm@labs.htt-consult.com

Xiaohu Xu
Huawei
Huawei Bld, No.156 Beiqing Rd.
Beijing, Hai-Dian District 100095
China

Email: xuxiaohu@huawei.com

Bingyang Liu
Huawei
Huawei Bld, No.156 Beiqing Rd.
Beijing, Hai-Dian District 100095
China

Email: xuxiaohu@huawei.com