Network Working Group
Internet-Draft
Intended status: Experimental
Expires: February 25, 2021

T. Herbert
Intel

August 24, 2020

Attribution Option for Extension Header Insertion
draft-herbert-6man-eh-attrib-02

Abstract

This document defines an "attribution option" that provides attribution for IPv6 extension headers, Hop-by-Hop options, or Destination options that are inserted by intermediate nodes in the delivery path of a packet. The purpose of this option is twofold: first it identifies the extension headers or options that have been inserted, secondly it attributes the inserted extension headers or options to the node responsible for inserting them.

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 https://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 February 25, 2021.

Table of Contents

1. Introduction

Extension header insertion has been proposed as a mechanism to annotate packets for transit across controlled, or limited domains ([I-D.voyer-6man-extension-header-insertion], [I-D.ietf-ippm-ioam-ipv6-options]). These annotations are in the form of inserted Hop-by-Hop or Destination options, or other inserted extension headers such Segment Routing Header. Presumably, before a packet egresses a controlled domain, any inserted extension headers or options should be removed.

Extension header insertion, removal, and other non-standard modifications at intermediate nodes are currently prohibited by

[RFC8200], and [I-D.smith-6man-in-flight-eh-insertion-harmful] provides the rationale for why extension header insertion is harmful and thus prohibited. This document addresses the main problem of extension header insertion which is the loss of attribution to the source of packet contents. An "attribution option", either as a Hop-by-Hop or Destination Option, is defined to provide proper attribution.

The attribution option provides two salient benefits:

* The attribution option unambiguously identifies what extension headers and Destination or Hop-by-Hop options were inserted by intermediate nodes.

* The attribution option includes an identification of the intermediate node that inserted extension headers or options into a packet.

1.1. Motivation for extension header insertion

IP-in-IP encapsulation has been proposed as an alternative to extension header insertion. While encapsulation may be functionally equivalent to header insertion, there are merits to header insertion:

* Extension header insertion can result in fewer bytes of overhead than encapsulation.

* The proper destination address to set in the encapsulating IP header may be unknown. For instance, a node might insert an extension header into an existing packet with the intent that the packet is routed based on the original destination to some egress node of the domain, and that node removes the inserted headers.

* Packets for a flow may require consistent routing whether or not extension headers are inserted. In particular, to route flows consistently in Equal Cost MultiPath (ECMP), the hash computed for ECMP should be the same for all packets of the flow. Unlike IP encapsulation, extension header insertion doesn't affect the fields used in ECMP hash calculation (the source address, destination address, flow label, and transport layer ports), so the ECMP hash calculation consistently derives the same value for all packets of a flow with or without inserted extension headers or options.

1.2. Problems with extension header and options insertion

Insertion or removal of extension headers, as well as Destination or Hop-by-Hop options, is currently prohibited by [RFC8200]:

> Extension headers (except for the Hop-by-Hop Options header) are not processed, inserted, or deleted by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header.

The rationale for this prohibition is articulated in [I-D.smith-6man-in-flight-eh-insertion-harmful]. A summary of cited problems with extension header and options insertion are:

* Extension header and options insertion break the attribution model of IP in that the contents of a packet are no longer attributable to the node identified by the source address of a packet (exceptions include data that a source sets in a packet that is explicitly specified to be modifiable).

* Extension header and options insertion break PMTU discovery since they increase the size of packets in flight.

* Extension header and options insertion breaks ICMP since inserted extension headers may themselves cause ICMP errors that are sent to the source address. If the source node receives such an ICMP error it cannot take any action to resolve the error since it's not the source of the data that caused the error.

* Extension header and options insertion may create a communications black hole if the data inserted by one node causes a packet to be dropped at a later downstream node. When this happens the source does not know the identity of the node that inserted the data and won't even know which node dropped the packet unless an ICMP error is sent. In any case, the sending host cannot address the issue hence persistent, systematic packet loss is possible. Such a scenario may be difficult to trouble shoot in an even moderately large network.

* Use of extension header insertion is generally assumed to be confined to a controlled domain where the domain is a walled garden such that inserted extension headers are always removed before packets would exit a domain. It is conceivable that configuration or implementation errors may allow packets with inserted extension headers to leak out of the controlled domain.

* Extension header and options insertion break the IP Authentication Header (AH) [RFC4302]. If a receiving node attempts to verify an authentication header that covers data inserted by intermediate nodes, then the packet authentication will fail and the packet will be dropped.

This proposal primarily addresses the attribution of packet contents problem. A solution to the attribution problem addresses or at least can mitigate the other problems with extension header insertion.

1.3. Inserting Hop-by-Hop options

For inserting Hop-by-Hop options into a packet there are two possibilities: 1) a Hop-by-Hop Options extension header already exists in the packet, 2) no Hop-by-Hop Options extension header exist in the packet so a Hop-by-Hop extension header is inserted into the packet which contains the options being inserted.

Note that per [RFC8200] there can only be one Hop-by-Hop Options extension header in a packet, and if present it must be the first extension header after the IPv6 header. If Hop-by-Hop Options are to be inserted into a packet with an existing Hop-by-Hop Options extension header, the options MUST be inserted into the options list for the existing extension header.

1.4. Inserting Destination options

Destination options may be inserted in Destination Options before or after the routing header. If an appropriate Destination Options extension header does not exist in the packet then a new Destination Options extension header containing the inserted options is inserted in the packet. The recommended ordering of extension headers in [RFC8200] SHOULD be maintained.

1.5. Inserting extension headers

When an extension header, not Hop-by-Hop or Destination, is inserted into a packet it is immediately preceded by a Destination Options extension header that includes an attribution option which describes the inserted extension header. If the extension header is being inserted immediately after an existing Destination Options extension header then the attribution option is inserted into the existing Destination Options extension header. If there is no preceding Destination Options extension header then one is created into which the attribution options is set.

1.6. Scope

This document describes a mechanism for providing attribution in extension header insertion and insertion of Hop-by-Hop and Destination Options. With the exception of inserting Hop-by-Hop Options and Destination Options, requirements and semantics for inserting specific types of extension headers are out of scope. Similarly, security aspects, including potential leakage of inserted headers outside of a controlled domain, is not in scope.

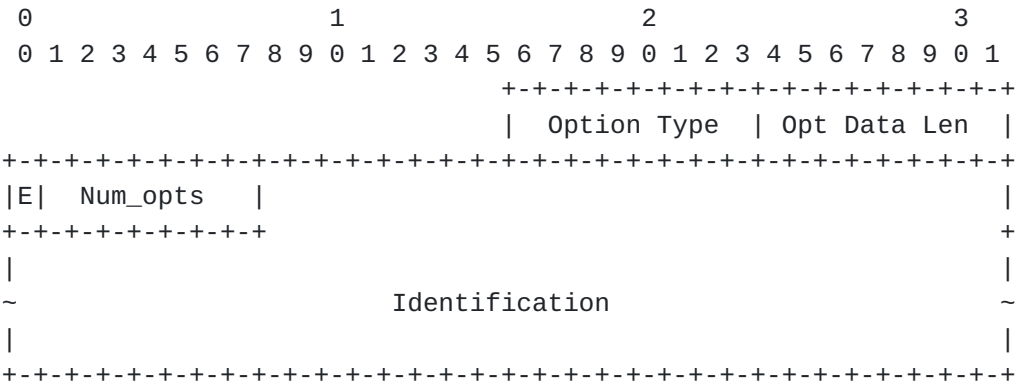
1.7. Requirements Language

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].

2. Attribution Option

2.1. Format

The format of the Hop-by-Hop or Destination Attribution Option is:



Fields are:

* Option Type: value is TBA. The first three bits of the option type should be 000 to indicate that the option is to be skipped over when processed as an unknown option and that the option data is unmodifiable.

* Opt Data Len: data length for the option. The minimal data length is one. If the data length equals twenty then the Identification is an IPv6 address (see section 2.1.2).

* E: For Destination Options this indicates that the extension header following the Destination Options extension header has

been inserted. When the option is in Hop-by-Hop Options, this bit MUST be zero when when transmitting and ignored on receive.

* Num_opts: If this value is less than 127 then it indicates the number of non-padding options following the Attribution Option that are attributed as being inserted. If the value is 127 then this indicates that the extension header was inserted and all following options are attributed as being inserted. Note that the maximum number of inserted options attributed by one Attribution Option is 126.

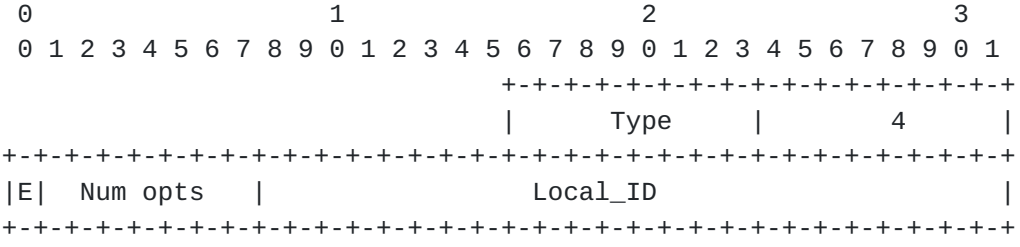
* Identification: indicates the source node responsible for the inserted extension headers. This can either be the IPv6 address of the responsible node or a local identifier value that is interpreted by the local network domain (see examples below). Note this field is variable length.

If options are being inserted into an existing Destination Options or Hop-by-Hop Options extension header then the Attribution Option is inserted as the first option in the header, followed by any inserted options, and then followed by any pre-existing options. The total length of the attribution option and any inserted options MUST be 8n; this ensures that any pre-existing options following those being inserted retain their original alignment. After the last inserted option the minimum amount of padding is added to make the total length of inserted data 8n. Pre-existing options, including padding, MUST NOT be modified other than moving them to follow the inserted options.

If a Destination or Hop-by-Hop Options extension header is being inserted in a packet then the Attribution Option is set as the first option in the header followed by an inserted options. Minimal padding MUST added make the length of the extension header 8n.

2.1.1. Attribution Option with short identifier

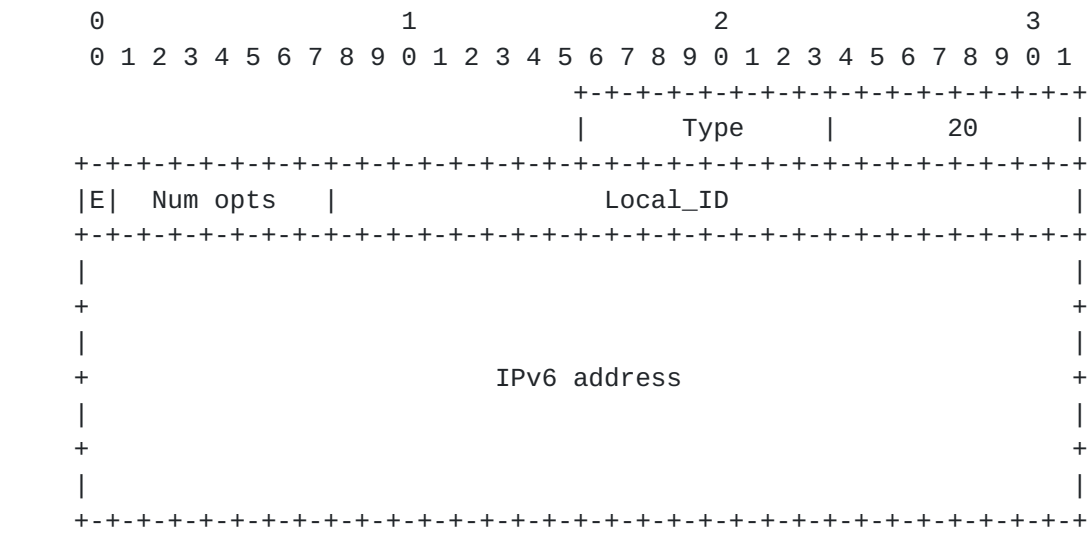
Below is the short format of the Attribution Option.



Local_ID is interpreted locally. For instance, it may be used as an index to a table to map a value to an IPv6 address.

2.1.2. Attribution Option with IPv6 address identifier

Below is the format of the Attribution Option that contains an IPv6 address for attribution of the inserted extension headers or options.



Local_ID contains supplemental identification that is interpreted by the local network. This MAY be the AS of network corresponding to the node identified by the IPv6 address.

2.2. Model

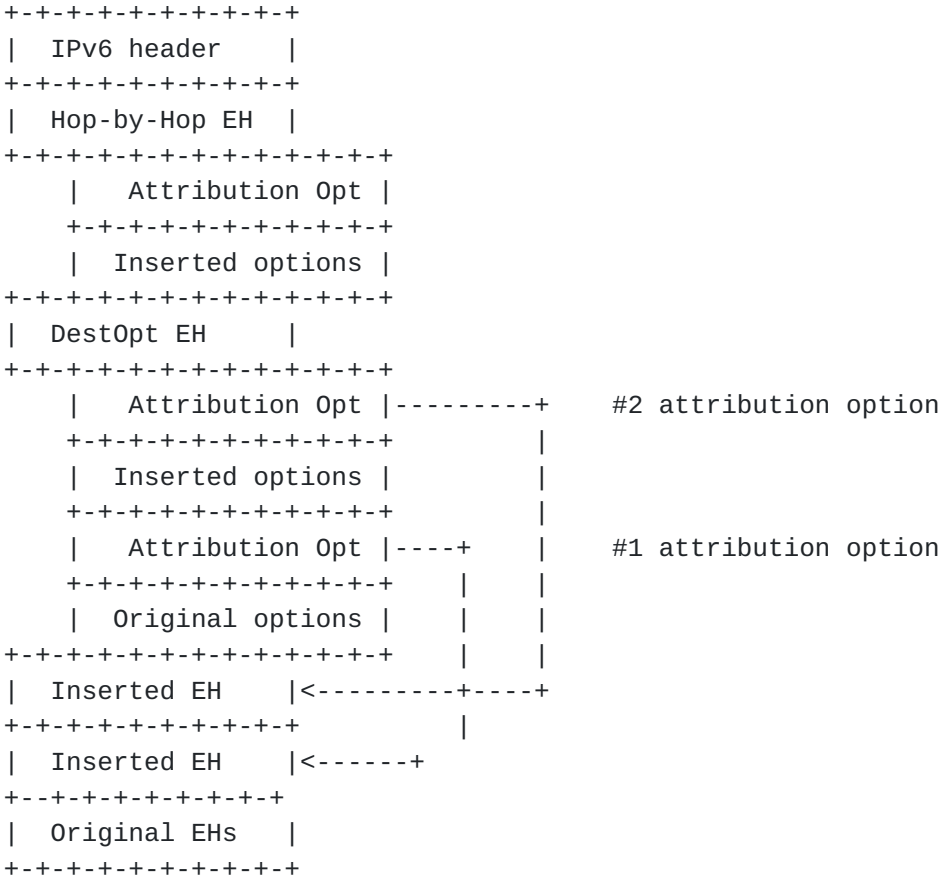
The Attribution Option indicates both inserted Hop-by-Hop or Destination options and inserted extension headers.

Multiple extension header or options insertions may occur during the lifetime of a packet. Insertions are treated as a stack. Hop-by-Hop and Destination options MUST be inserted in an extension header before any pre-existing options including those previously inserted. Similarly, if an extension header is being inserted and a corresponding attribution option is being added to a Destination Option extension header then the inserted extension header immediately follows the Destination Options extension header and precedes any previously inserted extension headers with an attribution option in the same Destination Options extension header.

Inserted extension headers and inserted Hop-by-Hop and Destination options MUST be removed in the reverse order of insertion (i.e. inserted headers are "popped" to remove them). When an Attribution Option is removed from a packet, which is the first option in the extension header, the option, any corresponding inserted options, and any inserted trailing padding are removed. In the case of a Destination Options or Hop-by-Hop Options extension header that was

inserted, the inserted extension header is removed when when the last attribution option in the extension header is removed (Num_opts in the option is equal to 127).

The logical structure of an IPv6 packet with inserted extension headers and options, and the relationship between Attribution Options and inserted extension headers and options, is demonstrated below. In this example, a Hop-by-Hop Options extension header was inserted that indicates inserted Hop-by-Hop options. There are two attribution options inserted into an existing Destination Options header: the first one (#1) indicates an inserted extension header and no options, the second (#2) indicates an inserted extension header and also inserted Destination options.



3. Operation

This section describes operations for extension header and options insertion and removal at intermediate nodes.

3.1. Insertion

An extension header or Hop-by-Hop or Destination options MAY be inserted into a packet. The packet's size will increase, and if options are inserted into Destination or Hop-by-Hop Options the size of those extension headers will increase.

3.1.1. Insertion procedure

Hop-by-Hop and Destination options, including the attribution option, are inserted into a packet with the following procedures.

Procedure is:

* If an appropriate Hop-by-Hop or Destination Options extension header does not exist in the packet:

  1) Insert a Hop-by-Hop or Destination Options extension header into the packet at the appropriate offset. The extension header contains the Attribution Option, followed by any Hop-by-Hop or Destination options being inserted. Num_opts is set to 127 to indicate that the extension header was inserted. E is set if another extension header is also being inserted (applicable to Destination Options). Add padding to make the length of the extension header be a multiple of eight bytes per [RFC8200].

  2) If no other extension header is being inserted then the nexthdr of the inserted Destination or Hop-by-Hop header is set to value of the nexthdr in the preceding IPv6 header or extension header.

  3) Else, if an extension header is being inserted then the nexthdr of the inserted Destination Options extension header is set to protocol number of the inserted extension header. The nexthdr for the inserted extension header is set to value of the original nexthdr in the IPv6 header or extension header that precedes the Destination Option being inserted.

  4) The nexthdr of the IPv6 header or extension header that precedes the inserted Destination of Hop-by-Hop Options is set to the protocol number for the inserted header (either 0 for Hop-by-Hop Options or 60 for Destination Options).

* Else, if an appropriate Hop-by-Hop or Destination Options extension header is already present then insert new options into the existing header:

1) Make first option to be the Attribution Option. Num_opts is set to the number of non-padding options being inserted not including the Attribution Option. E is set if an extension header is being inserted (applicable to Destination Options only).

2) Following the Attribution Option, set any other options being inserted. Include padding before the options as necessary to enforce any alignment requirements.

3) Following the last inserted option, add the minimal amount of padding such that the alignment of the first byte after the last inserted byte is 8n+2 from the start of the Hop-by-Hop or Destination extension header. This is necessary to preserve alignment requirements of existing options. The amount of padding needed is:

   7 - ((offset_last_inserted_byte - 3) % 8)

4) Following the last inserted option and inserted padding, copy the original options from the packet.

5) Set length of the Hop-by-Hop or Destination Options extension header to reflect the length with the inserted options and any inserted padding.

6) If an extension header is being inserted then the nexthdr of the Destination Options header is set to protocol number of the inserted extension header. The nexthdr for the inserted extension header is set to is set to original nexthdr value of Destination Options extension header.

3.1.2. Errors during insertion

Errors may occur in the process of inserting extension headers in a packet. Error conditions would include the resultant packet size exceeding MTU, and the size of Hop-by-Hop Options extension header exceeding 1024 bytes (the maximum size of the Hop-by-Hop Options extension header).

If an error occurs during insertion then the node performing insertion MUST take an appropriate behavior per some configuration. The packet MAY be discarded or the unmodified packet MAY be forwarded. An error SHOULD be logged.

3.2. Removal of inserted extension headers and options

The top level inserted extension headers and Hop-by-Hop or Destination options, referred to by the Attribution Option which is first option in the Hop-by-Hop or Destination options of a packet, MAY be removed by an intermediate node.

3.2.1. Removal procedure

The procedure is:

* If Num_opts equals 127 then the Destination or Hop-by-Hop extension header is to be removed.

  * If the E bit is not set or a Hop-by-Hop extension header is being removed, remove the Destination or Hop-by-Hop extension header bytes from the packet and set the nexthdr of the preceding IPv6 header or extension header to the nexthdr of the Destination or Hop-by-Hop Options extension header being removed.

  * Else, if the E bit is set in the attribution options of a Destination extension header, remove the extension header bytes of the following extension header from the packet. The nexthdr of the preceding IPv6 header or extension header is set to the nexthdr of the Hop-by-Hop Options extension header being removed.

* Else, if Num_opts is less than 127, then the inserted options must be removed from the existing header:

  1) Locate the last inserted option. This done by the scanning non-padding options after the Attribution Option for the count in Num_opts.

  2) Compute the amount of padding that was inserted. The amount of padding that should have been inserted is:

     7 - ((offset_last_inserted_byte - 2) % 8)

     where offset_last_byte is the offset of the last byte of the last inserted option located in step #1.

  3) Remove the bytes in the packet from first byte of the Destination or Hop-by-Hop Options data (first byte of the Attribution option) through the last byte of inserted padding as computed in step #2.

4) Set the length of the Hop-by-Hop Options extension header to account for the removed bytes; that is the orignal extension header length minus the number of removed bytes.

5 If the E bit is set in the attribution option being removed of a Destination extension header, remove the following extension header from the packet. The nexthdr of the Destination Options extension header is set to the nexthdr of the extension header being removed.

3.2.2. Errors during removal

A node performing extension header removal MUST validate packet contents.

The following attributes MUST be validated before removal:

* If Num_opts is not equal to 127 then number of non-padding options following Attribution Option MUST be greater than or equal to Num_opts.

* Necessary padding after the last inserted Hop-by-Hop option MUST be present. The amount of padding MUST be equal to the expected amount.

* The Num_opts options following the Attribution Option MUST NOT contain another Attribution Option.

* If the E bit is set in the Attribution options of a Destination Options header then the a valid extension header MUST follow the Destination Options header.

If any of the above validations fail, or an error is otherwise encountered in the removal process, then the processing node MUST take action. The packet SHOULD be discarded and error message SHOULD be logged.

3.3. Domain edge filtering

Filtering packets with inserted extension headers or Destination or Hop-by-Hop options is straightforward: a packet contains inserted options if the first option of a Destination Options or Hop-by-Hop Options is the Attribution Option. A packet contains inserted extension headers if it contains an attribution option, either in Destination Options or Hop-by-Hop Options, with Num_opts equal to 127; or it contains an attribution option in Destination Options that has the E bit set.

3.4. ICMP processing

At described in [I-D.smith-6man-in-flight-eh-insertion-harmful], it is possible for a source node to receive ICMP [RFC4443] errors caused by inserted headers, thus the source node has no recourse to address the error.

This section proposes some ways to apply the Attribution Option to mitigate the ICMP breakage for extension header insertion:

* ICMP errors can be filtered [RFC4890] by nodes in the network before reaching a source node outside of the domain (at the domain edge for instance). The packet headers in the ICMP data will include the Destination Options or Hop-by-Hop Options extension header containing the Attribution Option. The filtering node MAY analyze the error to determine if it was caused by the inserted headers:

  - If the error was caused by inserted extension headers, then the node SHOULD take appropriate actions (minimally it SHOULD log the error). The filtering node SHOULD not forward the ICMP error to the source.

  - If the error was not caused by inserted headers, the filtering node MAY create a new ICMP error with the data packet that would be reflect the packet contents prior to extension header insertion (i.e. attempt set the packet in ICMP to be that which the source would have sent). This is done by removing the inserted extension headers of the packet in the ICMP data, and adjusting the Pointer field in an ICMP error if necessary. The revised ICMP error can then be forwarded to the source.

* If ICMP errors are not filtered and the source node receives an ICMP error for a packet containing inserted extension headers:

  - If the source node is a legacy implementation that does not understand the Attribution Option then it will attempt to process the error under the assumption that it was the source of the packet and the data that caused the error. If the node logs the contents of the ICMP error, which should be common, then external out-of-band analysis can be done by network administrators to troubleshoot the ICMP errors and identify culprit if the error was caused by inserted extension headers.

  - If the source node understands the Attribution Option then it can perform more analysis. The node MAY attempt to

ascertain if the error was caused by inserted headers or not, and if not it can then attempt to fix the problem with the assumption the it was responsible for the data in error.

3.5. Processing AH

Extension headers and options MAY be inserted into a packet before an existing AH header. The inserted data is not covered in the ICV computation and if a receiving host attempts performs the ICV computation over inserted data it is expected that verification will fail and the packet will be dropped.

The simplest way to address this is to remove any inserted headers in the packet before processing the AH extension header. The assumption is that once the inserted data is removed the packet contents reflect the original contents set by the host so AH verification should succeed.

Host implementations can be modified to process the attribution option. When a packet with inserted headers or options is received by an end host, the AH processing can ignore any inserted Destination or Hop-by-Hop options and any inserted extension headers. This can be done in conjunction with the existing algorithms to ignore option data in the ICV computation for modifiable options. Effectively, the algorithm is simply to remove all the inserted options and extension headers following the procedures in section 3.1.

4. Security Considerations

The Attribution Option does not in itself introduce any new security considerations. The security of containing inserted extension headers within a controlled domain is out of scope for this document.

Section 3.5 describes the processing of the IP Authentication Header in the presence of inserted options or extension headers.

5. IANA Considerations

IANA is requested to assigned the following Destination and Hop-By-Hop option:

Hex Value	Binary value act chg rest	Description	Reference
TBD	00 0 TBD	Attribution Option	This document

6. References

6.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

6.2. Informative References

[I-D.ietf-ippm-ioam-ipv6-options] Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., and R. Asati, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-02 (work in progress), July 2020.

[I-D.smith-6man-in-flight-eh-insertion-harmful] Smith, M., Kottapalli, N., Bonica, R., Gont, F., and T. Herbert, "In-Flight IPv6 Extension Header Insertion Considered Harmful", draft-smith-6man-in-flight-eh-insertion-harmful-02 (work in progress), May 2020.

[I-D.voyer-6man-extension-header-insertion] Voyer, D., Filsfils, C., Dukes, D., Matsushima, S., Leddy, J., Li, Z., and J. Guichard, "Deployments With Insertion of IPv6 Segment Routing Headers", draft-voyer-6man-extension-header-insertion-09 (work in progress), May 2020.

[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <https://www.rfc-editor.org/info/rfc4890>.

Author's Address

Tom Herbert
Intel
Santa Clara, CA
USA

Email: tom@quantonium.net