ICMPv6 errors for discarding packets due to processing limits

draft-ietf-6man-icmp-limits-02

Abstract

Network nodes may discard packets if they are unable to process protocol headers of packets due to processing constraints or limits. When such packets are dropped, the sender receives no indication so it cannot take action to address the cause of discarded packets. This document defines ICMPv6 errors that can be sent by a node that discards packets because it is unable to process the protocol headers. A node that receives such an ICMPv6 error may be able to modify what it sends in future packets to avoid subsequent packet discards.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1 Introduction ........................................... 3
  1.1 Extension header limits .......................... 3
  1.2 Aggregate header limits ......................... 4
2 ICMPv6 errors for extension header limits ................. 4
  2.1 Format ........................................... 4
  2.2 Unrecognized Next Header type encountered (code 1) .... 5
  2.3 Extension header too big (code 4) ............... 5
  2.4 Extension header chain too long (code 5) .......... 6
  2.5 Too many options in extension header (code 6) ...... 6
  2.6 Option too big (code 7) ........................ 6
3 ICMPv6 error for aggregate header limits ................ 7
  3.1 Format ........................................... 7
  3.2 Usage ........................................... 8
4 Operation ............................................ 8
  4.1 Priority of reporting ............................ 8
  4.2 Host response ................................... 9
5 Security Considerations ................................. 10
6 IANA Considerations ................................ 11
  6.1 Parameter Problem codes .......................... 11
  6.2 Destination Unreachable codes .................... 11
7 Acknowledgments ...................................... 11
8 References ........................................... 11
  8.1 Normative References ............................. 11
  8.2 Informative References ........................... 12
Author’s Address ...................................... 12
1 Introduction

This document specifies ICMPv6 errors that can be sent when a node discards a packet due to it being unable to process the necessary protocol headers because of processing constraints or limits. New ICMPv6 code points are defined as an update to [RFC4443]. Five of the errors are specific to processing limits of extension headers; another error is used when the aggregate protocol headers in a packet exceed the processing limits of a node.

1.1 Extension header limits

In IPv6, optional internet-layer information is carried in one or more IPv6 Extension Headers [RFC8200]. Extension Headers are placed between the IPv6 header and the Upper-Layer Header in a packet. The term "Header Chain" refers collectively to the IPv6 header, Extension Headers, and Upper-Layer Headers occurring in a packet. Individual extension headers may have a length of 2048 octets and must fit into one MTU. Destination Options and Hop-by-Hop Options contain a list of options in Type-length-value (TLV) format. Each option includes a length of the data field in octets, and the minimum size of an option (non-pad type) is two bytes and the maximum length is 257 bytes. The number of options in an extension header is only limited by the length of the extension header and MTU. Options may also be skipped over by a receiver if they are unknown and the Option Type indicates to skip (first two high order bits are 00).

Per [RFC8200], except for Hop by Hop options, extension headers are not examined or processed by intermediate nodes. Many intermediate nodes, however, do examine extension header for various purposes. For instance, a node may examine all extension headers to locate the transport header of a packet in order to implement transport layer filtering or to track connections to implement a stateful firewall.

Destination hosts are expected to process all extensions headers and options in Hop-by-Hop and Destination Options.

Due to the variable lengths, high maximum lengths, or potential for Denial of Service attack of extension headers-- many devices impose operational limits on extension headers in packets they process. [RFC7045] discusses the requirements of intermediate nodes that discard packets because of unrecognized extension headers. [RFC8504] discusses limits that may be applied on the number of options in Hop-by-Hop or Destination Options extension headers. When a limit is exceeded, the typical behavior is to silently discard a packet. Both intermediate nodes and end hosts may institute limits on extension header processing.
This document defines four Parameter Problem codes and extends the applicability of an existing code that may be sent by a node that discards a packet due to processing limits of extension headers being exceeded. A source host that receives an ICMPv6 error can modify its use of extension headers in subsequent packets sent to the destination in order to avoid further occurrences of packets being discarded.

1.2 Aggregate header limits

Many hardware devices implement a parsing buffer of a fixed sized to process packets. The parsing buffer is expected to contain all the headers (often up to a transport layer header for filtering) that a device needs to examine. Parsing buffers have been implemented with various sizes (512 bytes is common, although some devices have smaller sizes). If the aggregate length of headers in a packet exceeds the size of the parsing buffer, a device will either discard the packet or defer processing to a software slow path. In any case, no indication of a problem is sent back to the sender.

This document defines one code for ICMPv6 Destination Unreachable that is sent by a node that is unable to process the headers of a packet due to the aggregate size of the packet headers exceeding a processing limit. A source host that receives an ICMPv6 error can modify the headers used in subsequent packets to try to avoid further occurrences of packets being discarded or relegated to a slow path.

2 ICMPv6 errors for extension header limits

Three new codes are defined for Parameter Problem type and applicability of one existing code is extended for ICMPv6 errors for extension header limits.

2.1 Format

The format of the ICMPv6 message for an extension header limit exceeded error is:

```
0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Pointer                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    As much of invoking packet                 |
|               as possible without the ICMPv6 packet           |
|               + exceeding the minimum IPv6 MTU [IPv6]        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

T. Herbert             Expires November 24, 2019
IPv6 Fields:

Destination Address
Copied from the Source Address field of the invoking packet.

ICMPv6 Fields:

Type
  4 (Parameter Problem type)

Code (pertinent to this specification)
  1 - Unrecognized Next Header type encountered
  4 - Extension header too big
  5 - Extension header chain too long
  6 - Too many options in extension header
  7 - Option too big

Pointer
Identifies the octet offset within the invoking packet where
the problem occurred.

The pointer will point beyond the end of the ICMPv6 packet if
the field having a problem is beyond what can fit in the
maximum size of an ICMPv6 error message.

2.2 Unrecognized Next Header type encountered (code 1)

[RFC8200] specifies that a destination host should send an
"unrecognized next header type" when a Next Header value is
unrecognized in a packet. This document extends this to allow
intermediate nodes to send this same error for a packet that is
discarded because the node does not recognize a Next Header type.

This code SHOULD be sent by an intermediate node that discards a
packet because it encounters a Next Header type that is unknown in
its examination. The ICMPv6 Pointer field is set to the offset of the
unrecognized value within the original packet.

Note that when the original sender receives the ICMPv6 error it can
differentiate between the message being sent by a destination host,
per [RFC4443], and an error sent by an intermediate host based on
matching the source address of the ICMPv6 packet and the destination
address of the packet in the ICMPv6 data.

2.3 Extension header too big (code 4)

An ICMPv6 Parameter Problem with code for "extension header too big"
SHOULD be sent when a node discards a packet because the size of an
extension header exceeds its processing limit. The ICMPv6 Pointer field is set to the offset of the first octet in the extension header that exceeds the limit.

2.4 Extension header chain too long (code 5)

An ICMPv6 Parameter Problem with code for "extension header chain too long" SHOULD be sent when a node discards a packet with an extension header chain because an extension header chain exceeds it processing limit.

There are two different limits that might be applied: a limit on the total size in octets of the header chain, and a limit on the number of extension headers in the chain. This error code is used in both cases. In the case that the size limit is exceeded, the ICMPv6 Pointer is set to the first octet beyond the limit. In the case that the number of extension headers is exceeded, the ICMPv6 Pointer is set to the offset of the first octet of the first extension header that is beyond the limit.

2.5 Too many options in extension header (code 6)

An ICMPv6 Parameter Problem with code for "too many options in extension header" SHOULD be sent when a node discards a packet with an extension header that has a number of options that exceed the processing limits of the node. This code is applicable for Destination options and Hop-by-Hop options. The ICMPv6 Pointer field is set to the first octet of the first option that exceeds the limit.

2.6 Option too big (code 7)

An ICMPv6 Parameter Problem with code for "option too big" is sent in two different cases: when the length of an individual option exceeds a limit, or when the length or number of consecutive padding options exceeds a limit.

If a packet is discarded because the length of a Hop-by-Hop or Destination option exceeds a processing limit, a node SHOULD send an ICMPv6 Parameter Problem with code equal to 7. The ICMPv6 Pointer field is set to the offset of the first octet in the option that exceeds the limit.

If a packet is discarded because the length or number of consecutive padding options (PAD1 and PADN) exceeds a limit, a node SHOULD send an ICMPv6 Parameter Problem with code equal to 7. The ICMPv6 Pointer field is set to the offset of first octet of the padding option that exceeds the limit.
Possible limits related to padding include:

* The number of consecutive PAD1 options in destination options or hop-by-hop options is limited to seven octets [RFC8504].

* The length of a PADN options in destination options or hop-by-hop options is limited seven octets [RFC8504].

* The aggregate length of a set of consecutive PAD1 or PADN options in destination options or hop-by-hop options is limited to seven octets.

3 ICMPv6 error for aggregate header limits

One code is defined for Destination Unreachable type for aggregate header limits.

3.1 Format

The error for aggregate header limits employs a multi-part ICMPv6 message format as defined in [RFC4884]. The extended structure contains a pointer to the octet beyond the limit.

The format of the ICMPv6 message for an aggregate header limit exceeded is:

```
  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     unused    |    Length     |           unused              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Internet Header + leading octets of original datagram |
|                                                               |
|                           //                                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Pointer                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
IPv6 Fields:

Destination Address
Copied from the Source Address field of the invoking packet.
ICMPv6 Fields:

Type
  1 (Destination Unreachable type)

Code (pertinent to this specification)
  8 - Headers too long

Length
  Length of the "original datagram" measured in 64 bit words

Pointer
  Identifies the octet offset within the invoking packet where a limit was exceeded.

The pointer will point beyond the end of the original datagram if the field exceeding the limit is beyond what can fit in the maximum size of an ICMPv6 error message.

3.2 Usage

An ICMPv6 Destination Unreachable error with code for "headers too long" SHOULD be sent when a node discards a packet because the aggregate length of headers in the packet exceeds the processing limits of the node. The Pointer in the extended ICMPv6 structure is set to the offset of the first octet that exceeds the limit.

4 Operation

Nodes that send or receive ICMPv6 errors due to header processing limits MUST generally comply with ICMPv6 processing as specified in [RFC4443].

4.1 Priority of reporting

More than one ICMPv6 error may be applicable to report for a packet. For instance, the number of extension headers in a packet might exceed a limit and the aggregate length of protocol headers might also exceed a limit. Only one ICMPv6 error SHOULD be sent for a packet, so a priority is defined to determine which error to report.

The RECOMMENDED reporting priority of ICMPv6 errors for processing limits is from highest to lowest priority:

1) Real error (existing codes)

2) "Unrecognized Next Header type" encountered by an intermediate node
3) "Extension header too big"

4) "Option too big" for length or number of consecutive padding options exceeding a limit

5) "Option too big" for the length of an option exceeding a limit

6) "Too many options in an extension header"

7) "Extension header chain too long" for number of extension headers exceeding a limit

8) "Extension header chain too long" for size of the extension header chain exceeding a limit

9) "Headers too long"

4.2 Host response

When a source host receives an ICMPv6 error for a processing limit being exceeded, it SHOULD verify the ICMPv6 error is valid and take an appropriate action.

The ICMPv6 error SHOULD be logged with sufficient detail for debugging packet loss. The details of the error, including the addresses and the offending extension header or data, should be retained. This, for instance, would be useful for debugging when a node is mis-configured and unexpectedly discarding packets, or when a new extension header is being deployed.

A host MAY modify its usage of protocol headers in subsequent packets to avoid repeated occurrences of the same error.

For ICMPv6 errors caused by extension header limits being exceeded:

* An error SHOULD be reported to an application if the application enabled extension headers for its traffic. In response, the application MAY terminate communications if extension headers are required, stop using extension headers in packets to the destination indicated by the ICMPv6 error, or attempt modify its use of extension headers or headers to avoid further packet discards.

* A host system SHOULD take appropriate action if it is automatically inserting extension headers into packets on behalf of the application. If the offending extension header is not required for communication, the host MAY either stop sending it or otherwise modify its use in subsequent packets sent to the...
destination indicated in the ICMPv6 error.

5 Security Considerations

This document does not introduce any new security concerns for use of ICMPv6 errors. The security considerations for ICMPv6 described in [RFC4443] are applicable.
6 IANA Considerations

6.1 Parameter Problem codes

IANA is requested to assign the following codes for ICMPv6 type 4 "Parameter Problem":

4 - Extension header too big
5 - Extension header chain too long
6 - Too many options in extension header
7 - Option too big

6.2 Destination Unreachable codes

IANA is requested to assign the following codes for ICMPv6 type 1 "Destination Unreachable":

8 - Headers too long

7 Acknowledgments

The author would like to thank Ron Bonica and Bob Hinden for their comments and suggestions that improved this document.

8 References

8.1 Normative References


[ RFC4884 ] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884,
8.2 Informative References


Author’s Address

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@quantonium.net