

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: [2460](#), [6145](#) (if approved)
Intended status: Standards Track
Expires: January 5, 2016

F. Gont
SI6 Networks / UTN-FRH
W. Liu
Huawei Technologies
T. Anderson
Redpill Linpro
July 4, 2015

Deprecating the Generation of IPv6 Atomic Fragments
draft-ietf-6man-deprecate-atomfrag-generation-02

Abstract

The core IPv6 specification requires that when a host receives an ICMPv6 "Packet Too Big" message reporting an MTU smaller than 1280 bytes, the host includes a Fragment Header in all subsequent packets sent to that destination, without reducing the assumed Path-MTU. The simplicity with which ICMPv6 "Packet Too Big" messages can be forged, coupled with the widespread filtering of IPv6 fragments, results in an attack vector that can be leveraged for Denial of Service purposes. This document briefly discusses the aforementioned attack vector, and formally updates [RFC2460](#) such that generation of IPv6 atomic fragments is deprecated, thus eliminating the aforementioned attack vector.

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 January 5, 2016.

Copyright Notice

Copyright (c) 2015 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	2
2.	Terminology	3
3.	Denial of Service (DoS) attack vector	3
4.	Additional Considerations	5
5.	Updating RFC2460	6
6.	IANA Considerations	7
7.	Security Considerations	7
8.	Acknowledgements	7
9.	References	7
9.1.	Normative References	7
9.2.	Informative References	8
Appendix A.	Small Survey of OSeS that Fail to Produce IPv6 Atomic Fragments	9
	Authors' Addresses	9

[1.](#) Introduction

[RFC2460] specifies the IPv6 fragmentation mechanism, which allows IPv6 packets to be fragmented into smaller pieces such that they fit in the Path-MTU to the intended destination(s).

[Section 5 of \[RFC2460\]](#) states that, when a host receives an ICMPv6 "Packet Too Big" message [[RFC4443](#)] advertising an MTU smaller than 1280 bytes (the minimum IPv6 MTU), the host is not required to reduce the assumed Path-MTU, but must simply include a Fragment Header in all subsequent packets sent to that destination. The resulting packets will thus *not* be actually fragmented into several pieces, but rather just include a Fragment Header with both the "Fragment Offset" and the "M" flag set to 0 (we refer to these packets as "atomic fragments"). As required by [[RFC6946](#)], these atomic fragments are essentially processed by the destination host as non-

fragment traffic (since there are not really any fragments to be reassembled). The goal of these atomic fragments has been to convey an appropriate Fragment Identification value to be employed by IPv6/IPv4 translators for the resulting IPv4 fragments.

While atomic fragments might seem rather benign, there are scenarios in which the generation of IPv6 atomic fragments can introduce an attack vector that can be exploited for denial of service purposes. Since there are concrete security implications arising from the generation of IPv6 atomic fragments, and there is no real gain in generating IPv6 atomic fragments (as opposed to e.g. having IPv6/IPv4 translators generate a Fragment Identification value themselves), this document formally updates [\[RFC2460\]](#), forbidding the generation of IPv6 atomic fragments, such that the aforementioned attack vector is eliminated.

[Section 3](#) describes some possible attack scenarios. [Section 4](#) provides additional considerations regarding the usefulness of generating IPv6 atomic fragments. [Section 5](#) formally updates [RFC2460](#) such that this attack vector is eliminated.

[2.](#) Terminology

IPv6 atomic fragments

IPv6 packets that contain a Fragment Header with the Fragment Offset set to 0 and the M flag set to 0 (as defined by [\[RFC6946\]](#)).

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.](#) Denial of Service (DoS) attack vector

Let us assume that Host A is communicating with Server B, and that, as a result of the widespread filtering of IPv6 packets with extension headers (including fragmentation) [\[I-D.ietf-v6ops-ipv6-ehs-in-real-world\]](#), some intermediate node filters fragments between Host A and Server B. If an attacker sends a forged ICMPv6 "Packet Too Big" (PTB) error message to server B, reporting an MTU smaller than 1280, this will trigger the generation of IPv6 atomic fragments from that moment on (as required by

[RFC2460]). When server B starts sending IPv6 atomic fragments (in response to the received ICMPv6 PTB), these packets will be dropped, since we previously noted that packets with IPv6 EHs were being dropped between Host A and Server B. Thus, this situation will result in a Denial of Service (DoS) scenario.

Another possible scenario is that in which two BGP peers are employing IPv6 transport, and they implement ACLs to drop IPv6 fragments (to avoid control-plane attacks). If the aforementioned BGP peers drop IPv6 fragments but still honor received ICMPv6 Packet Too Big error messages, an attacker could easily attack the peering session by simply sending an ICMPv6 PTB message with a reported MTU

smaller than 1280 bytes. Once the attack packet has been sent, it will be the aforementioned routers themselves the ones dropping their own traffic.

The aforementioned attack vector is exacerbated by the following factors:

- o The attacker does not need to forge the IPv6 Source Address of his attack packets. Hence, deployment of simple [BCP38](#) filters will not help as a counter-measure.
- o Only the IPv6 addresses of the IPv6 packet embedded in the ICMPv6 payload needs to be forged. While one could envision filtering devices enforcing [BCP38](#)-style filters on the ICMPv6 payload, the use of extension headers (by the attacker) could make this difficult, if at all possible.
- o Many implementations fail to perform validation checks on the received ICMPv6 error messages, as recommended in [Section 5.2 of RFC4443](#) and documented in [RFC5927](#). It should be noted that in some cases, such as when an ICMPv6 error message has (supposedly) been elicited by a connection-less transport protocol (or some other connection-less protocol being encapsulated in IPv6), it may be virtually impossible to perform validation checks on the received ICMPv6 error messages. And, because of IPv6 extension headers, the ICMPv6 payload might not even contain any useful information on which to perform validation checks.
- o Upon receipt of one of the aforementioned ICMPv6 "Packet Too Big"

error messages, the Destination Cache [[RFC4861](#)] is usually updated to reflect that any subsequent packets to such destination should include a Fragment Header. This means that a single ICMPv6 "Packet Too Big" error message might affect multiple communication instances (e.g., TCP connections) with such destination.

- o As noted in [Section 4](#), SIIT [[RFC6145](#)] (including derivative protocols such as Stateful NAT64 [[RFC6146](#)]) is the only technology which currently makes use of atomic fragments. Unfortunately, an IPv6 node cannot easily limit its exposure to the aforementioned attack vector by only generating IPv6 atomic fragments towards IPv4 destinations behind a stateless translator. This is due to the fact that [Section 3.3 of RFC6052](#) [[RFC6052](#)] encourages operators to use a Network-Specific Prefix (NSP) that maps the IPv4 address space into IPv6. When an NSP is being used, IPv6 addresses representing IPv4 nodes (reached through a stateless translator) are indistinguishable from native IPv6 addresses.

[4.](#) Additional Considerations

Besides the security assessment provided in [Section 3](#), it is interesting to evaluate the pros and cons of having an IPv6-to-IPv4 translating router rely on the generation of IPv6 atomic fragments.

Relying on the generation of IPv6 atomic fragments implies a reliance on:

1. ICMPv6 packets arriving from the translator to the IPv6 node
2. The ability of the nodes receiving ICMPv6 PTB messages reporting an MTU smaller than 1280 bytes to actually produce atomic fragments
3. Support for IPv6 fragmentation on the IPv6 side of the translator

Unfortunately,

- o There exists a fair share of evidence of ICMPv6 Packet Too Big messages being dropped on the public Internet (for instance, that is one of the reasons for which PLPMTUD [[RFC4821](#)] was produced).

Therefore, relying on such messages being successfully delivered will affect the robustness of the protocol that relies on them.

- o A number of IPv6 implementations have been known to fail to generate IPv6 atomic fragments in response to ICMPv6 PTB messages reporting an MTU smaller than 1280 bytes (see [Appendix A](#) for a small survey). Additionally, the results included in [Section 6 of \[RFC6145\]](#) note that 57% of the tested web servers failed to produce IPv6 atomic fragments in response to ICMPv6 PTB messages reporting an MTU smaller than 1280 bytes. Thus, any protocol relying on IPv6 atomic fragment generation for proper functioning will have interoperability problems with the aforementioned IPv6 stacks.
- o IPv6 atomic fragment generation represents a case in which fragmented traffic is produced where otherwise it would not be needed. Since there is widespread filtering of IPv6 fragments in the public Internet [[I-D.ietf-v6ops-ipv6-ehs-in-real-world](#)], this would mean that the (unnecessary) use of IPv6 fragmentation might result, unnecessarily, in a Denial of Service situation even in legitimate cases.

Finally, we note that SIIT essentially employs the Fragment Header of IPv6 atomic fragments to signal the translator how to set the DF bit of IPv4 datagrams (the DF bit is cleared when the IPv6 packet contains a Fragment Header, and is otherwise set to 1 when the IPv6

packet does not contain an IPv6 Fragment Header). Additionally, the translator will employ the low-order 16-bits of the IPv6 Fragment Identification for setting the IPv4 Fragment Identification. At least in theory, this is expected to reduce the Fragment ID collision rate in the following specific scenario:

1. An IPv6 node communicates with an IPv4 node (through SIIT)
2. The IPv4 node is located behind an IPv4 link with an MTU < 1260
3. ECMP routing [[RFC2992](#)] with more than one translator is employed for e.g., redundancy purposes

In such a scenario, if each translator were to select the IPv4 Fragment Identification on its own (rather than selecting the IPv4

Fragment ID from the low-order 16-bits of the Fragment Identification of atomic fragments), this could possibly lead to IPv4 Fragment ID collisions. However, since a number of implementations set IPv6 Fragment ID according to the output of a Pseudo-Random Number Generator (PRNG) (see [Appendix B](#) of [\[I-D.ietf-6man-predictable-fragment-id\]](#)) and the translator only employs the low-order 16-bits of such value, it is very unlikely that relying on the Fragment ID of the IPv6 atomic fragment will result in a reduced Fragment ID collision rate (when compared to the case where the translator selects each IPv4 Fragment ID on its own).

Finally, we note that [\[RFC6145\]](#) is currently the only "consumer" of IPv6 atomic fragments, and it correctly and diligently notes (in [Section 6](#)) the possible interoperability problems of relying on IPv6 atomic fragments, proposing as a workaround that leads to more robust behavior and simplified code.

5. Updating [RFC2460](#)

The following text from [Section 5 of \[RFC2460\]](#):

"In response to an IPv6 packet that is sent to an IPv4 destination (i.e., a packet that undergoes translation from IPv6 to IPv4), the originating IPv6 node may receive an ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280. In that case, the IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment header in those packets so that the IPv6-to-IPv4 translating router can obtain a suitable Identification value to use in resulting IPv4 fragments. Note that this means the payload may have to be reduced to 1232 octets (1280 minus 40 for the IPv6 header and 8 for the Fragment header), and smaller still if additional extension headers are used."

is formally replaced with:

"An IPv6 node that receives an ICMPv6 Packet Too Big error message that reports a Next-Hop MTU smaller than 1280 bytes (the minimum IPv6 MTU) MUST NOT include a Fragment header in subsequent packets sent to the corresponding destination. That is, IPv6 nodes MUST NOT generate IPv6 atomic fragments."

6. IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

7. Security Considerations

This document describes a Denial of Service (DoS) attack vector that leverages the widespread filtering of IPv6 fragments in the public Internet by means of ICMPv6 PTB error messages. Additionally, it formally updates [[RFC2460](#)] such that this attack vector is eliminated.

8. Acknowledgements

The authors would like to thank (in alphabetical order) Alberto Leiva, Bob Briscoe, Brian Carpenter, Tatuya Jinmei, Jeroen Massar, and Erik Nordmark, for providing valuable comments on earlier versions of this document.

Fernando Gont would like to thank Fernando Gont would like to thank Jan Zorz / Go6 Lab <<http://go6lab.si/>>, and Jared Mauch / NTT America, for providing access to systems and networks that were employed to produce some of tests that resulted in the publication of this document. Additionally, he would like to thank SixXS <<https://www.sixxs.net>> for providing IPv6 connectivity.

9. References

9.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 4443](#), March 2006.

- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", [RFC 4821](#), March 2007.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", [RFC 6145](#), April 2011.

9.2. Informative References

- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", [RFC 2923](#), September 2000.
- [RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", [RFC 2992](#), November 2000.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", [RFC 5927](#), July 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", [RFC 6052](#), October 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", [RFC 6146](#), April 2011.
- [RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", [RFC 6946](#), May 2013.
- [I-D.ietf-6man-predictable-fragment-id]
Gont, F., "Security Implications of Predictable Fragment Identification Values", [draft-ietf-6man-predictable-fragment-id-08](#) (work in progress), June 2015.
- [I-D.ietf-v6ops-ipv6-ehs-in-real-world]
Gont, F., Linkova, J., Chown, T., and S. LIU, "Observations on IPv6 EH Filtering in the Real World", [draft-ietf-v6ops-ipv6-ehs-in-real-world-00](#) (work in progress), April 2015.

[Morbitzer]

Morbitzer, M., "TCP Idle Scans in IPv6", Master's Thesis. Thesis number: 670. Department of Computing Science, Radboud University Nijmegen. August 2013, <https://www.ru.nl/publish/pages/578936/m_morbitzer_masterthesis.pdf>.

Appendix A. Small Survey of Oses that Fail to Produce IPv6 Atomic Fragments

[This section will probably be removed from this document before it is published as an RFC].

This section includes a non-exhaustive list of operating systems that **fail** to produce IPv6 atomic fragments. It is based on the results published in [RFC6946] and [Morbitzer].

The following Operating Systems fail to generate IPv6 atomic fragments in response to ICMPv6 PTB messages that report an MTU smaller than 1280 bytes:

- o FreeBSD 8.0
- o Linux kernel 2.6.32
- o Linux kernel 3.2
- o Mac OS X 10.6.7
- o NetBSD 5.1

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Internet-Draft Deprecate Generation of IPv6 Atomic Frags

July 2015

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

Tore Anderson
Redpill Linpro
Vitaminveien 1A
Oslo 0485
Norway

Phone: +47 959 31 212

Email: tore@redpill-linpro.com

URI: <http://www.redpill-linpro.com>

Gont, et al.

Expires January 5, 2016

[Page 10]