Internet Engineering Task Force (IETF)
Request for Comments: 6946
Updates: 2460, 5722
Category: Standards Track
ISSN: 2070-1721

F. Gont
Huawei Technologies

May 2013

Processing of IPv6 "Atomic" Fragments

Abstract

The IPv6 specification allows packets to contain a Fragment Header without the packet being actually fragmented into multiple pieces (we refer to these packets as "atomic fragments"). Such packets are typically sent by hosts that have received an ICMPv6 "Packet Too Big" error message that advertises a Next-Hop MTU smaller than 1280 bytes, and are currently processed by some implementations as normal "fragmented traffic" (i.e., they are "reassembled" with any other queued fragments that supposedly correspond to the same original packet). Thus, an attacker can cause hosts to employ atomic fragments by forging ICMPv6 "Packet Too Big" error messages, and then launch any fragmentation-based attacks against such traffic. This document discusses the generation of the aforementioned atomic fragments and the corresponding security implications. Additionally, this document formally updates RFC 2460 and RFC 5722, such that IPv6 atomic fragments are processed independently of any other fragments, thus completely eliminating the aforementioned attack vector.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6946.

Table of Contents

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). [RFC2460] allows fragments to overlap, thus leading to ambiguity in the result of the reassembly process, which could be leveraged by attackers to bypass firewall rules and/or evade Network Intrusion Detection Systems (NIDS) [RFC5722].

[RFC5722] forbids overlapping fragments, specifying that when overlapping fragments are detected, all the fragments corresponding to that packet must be silently discarded.

As specified in Section 5 of [RFC2460], when a host receives an ICMPv6 "Packet Too Big" message advertising a "Next-Hop MTU" smaller than 1280 (the minimum IPv6 MTU), it 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 actually be fragmented into several pieces but will 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"). IPv6/IPv4 translators employ the Fragment Identification information found in the Fragment Header to select an appropriate Fragment Identification value for the resulting IPv4 fragments.

While these packets are really atomic fragments (they can be processed by the IPv6 module and handed to the upper-layer protocol without waiting for any other fragments), many IPv6 implementations process them as regular fragments. Namely, they try to perform IPv6 fragment reassembly with the atomic fragment and any other fragments already queued with the same set {IPv6 Source Address, IPv6 Destination Address, Fragment Identification}. For example, in the case of IPv6 implementations that have been updated to support [RFC5722], if a fragment with the same {IPv6 Source Address, IPv6 Destination Address, Fragment Identification} is already queued for reassembly at a host when an atomic fragment is received with the same set {IPv6 Source Address, IPv6 Destination Address, Fragment Identification}, and both fragments overlap, all the fragments will be silently discarded.

Processing of IPv6 atomic fragments as regular fragmented packets clearly provides an unnecessary vector to perform fragmentation-based attacks against non-fragmented traffic (i.e., IPv6 datagrams that are not really split into multiple pieces but that just include a Fragment Header).

IPv6 fragmentation attacks have been discussed in great detail in [PREDICTABLE-ID] and [CPNI-IPv6], and [RFC5722] describes a specific firewall-circumvention attack that could be performed by leveraging overlapping fragments. The possible IPv6 fragmentation-based attacks are, in most cases, "ports" of the IPv4 fragmentation attacks discussed in [RFC6274].

Section 3 describes the generation of IPv6 atomic fragments and how they can be remotely "triggered" by a remote attacker. Section 4 formally updates [RFC2460] and [RFC5722] such that the aforementioned attack vector is eliminated. Appendix A contains a survey of the generation and processing of IPv6 atomic fragments in different versions of a number of popular IPv6 implementations.

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.

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. Generation of IPv6 Atomic Fragments

Section 5 of [RFC2460] states:

> "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."

This means that any ICMPv6 "Packet Too Big" message advertising a "Next-Hop MTU" smaller than 1280 could trigger the generation of the so-called "atomic fragments" (i.e., IPv6 datagrams that include a Fragment Header but that are composed of a single fragment, with both the "Fragment Offset" and the "M" fields of the Fragment Header set to 0). This can be leveraged to perform a variety of fragmentation-based attacks [PREDICTABLE-ID] [CPNI-IPv6].

> For example, an attacker could forge IPv6 fragments with an appropriate {IPv6 Source Address, IPv6 Destination Address, Fragment Identification} tuple, such that these malicious fragments are incorrectly "reassembled" by the destination host together with some of the legitimate fragments of the original packet, thus leading to packet drops (and a potential denial of service).

From a security standpoint, this situation is exacerbated by the following factors:

- 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 connectionless transport protocol (or some other connectionless protocol being encapsulated in IPv6), it may be virtually impossible to perform validation checks on the received ICMPv6 error messages.

- 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 that 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 that IPv6 destination address.

- Some implementations employ predictable Fragment Identification values, thus greatly improving the chances of an attacker successfully performing fragmentation-based attacks [PREDICTABLE-ID].

4. Updating RFC 2460 and RFC 5722

Section 4.5 of [RFC2460] and Section 4 of [RFC5722] are updated as follows:

> A host that receives an IPv6 packet that includes a Fragment Header with the "Fragment Offset" equal to 0 and the "M" flag equal to 0 MUST process that packet in isolation from any other packets/fragments, even if such packets/fragments contain the same set {IPv6 Source Address, IPv6 Destination Address, Fragment Identification}. A received atomic fragment should be "reassembled" from the contents of that sole fragment.
>
> > The Unfragmentable Part of the reassembled packet consists of all headers up to, but not including, the Fragment Header of the received atomic fragment.
> >
> > The Next Header field of the last header of the Unfragmentable Part of the reassembled packet is obtained from the Next Header field of the Fragment Header of the received atomic fragment.

The Payload Length of the reassembled packet is obtained by subtracting the length of the Fragment Header (that is, 8) from the Payload Length of the received atomic fragment.

Additionally, if any fragments with the same set {IPv6 Source Address, IPv6 Destination Address, Fragment Identification} are present in the fragment reassembly queue when the atomic fragment is received, such fragments MUST NOT be discarded upon receipt of the "colliding" IPv6 atomic fragment, since IPv6 atomic fragments MUST NOT interfere with "normal" fragmented traffic.

5. Security Considerations

This document describes how the traditional processing of IPv6 atomic fragments enables the exploitation of fragmentation-based attacks (such as those described in [PREDICTABLE-ID] and [CPNI-IPv6]). This document formally updates [RFC2460] and [RFC5722], such that IPv6 atomic fragments are processed independently of any other fragments, thus completely eliminating the aforementioned attack vector.

6. Acknowledgements

The author would like to thank (in alphabetical order) Tore Anderson, Ran Atkinson, Remi Despres, Stephen Farrell, Brian Haberman, Timothy Hartrick, Steinar Haug, Philip Homburg, Simon Josefsson, Simon Perreault, Sean Turner, Florian Weimer, and Bjoern A. Zeeb for providing valuable comments on earlier versions of this document. Additionally, the author would like to thank Alexander Bluhm, who implemented this specification for OpenBSD.

This document is based on the technical report "Security Assessment of the Internet Protocol version 6 (IPv6)" [CPNI-IPv6], authored by Fernando Gont on behalf of the UK Centre for the Protection of National Infrastructure (CPNI).

Finally, the author wishes to thank Nelida Garcia and Guillermo Gont for their love and support.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, December 2009.

7.2. Informative References

[CPNI-IPv6] Gont, F., "Security Assessment of the Internet Protocol version 6 (IPv6)", UK Centre for the Protection of National Infrastructure, (available on request).

[PREDICTABLE-ID] Gont, F., "Security Implications of Predictable Fragment Identification Values", Work in Progress, March 2013.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, July 2010.

[RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, July 2011.

Appendix A. Survey of Processing of IPv6 Atomic Fragments by Different Operating Systems

This section includes a survey of the support of IPv6 atomic fragments in popular operating systems, as tested on October 30, 2012.

Operating System	Generates atomic fragments	Implements this specification
FreeBSD 8.0	No	No
FreeBSD 8.2	Yes	No
FreeBSD 9.0	Yes	No
Linux 3.0.0-15	Yes	Yes
NetBSD 5.1	No	No
NetBSD-current	No	Yes
OpenBSD-current	Yes	Yes
Solaris 11	Yes	Yes
Windows XP SP2	Yes	No
Windows Vista (Build 6000)	Yes	No
Windows 7 Home Premium	Yes	No

Table 1: Processing of IPv6 Atomic Fragments by Different OSes

In the table above, "generates atomic fragments" notes whether an implementation generates atomic fragments in response to received ICMPv6 "Packet Too Big" error messages that advertise an MTU smaller than 1280 bytes.

Author's Address

Fernando Gont
Huawei Technologies
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
EMail: fgont@si6networks.com