IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 2460, 5722 (if approved)
Intended status: Standards Track
Expires: August 4, 2012

F. Gont
UK CPNI

February 1, 2012

# Processing of IPv6 "atomic" fragments
# draft-ietf-6man-ipv6-atomic-fragments-00

Abstract

The IPv6 specification allows packets to contain a Fragment Header without the packet being actually fragmented into multiple pieces. Such packets typically result from 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 "fragmented traffic". Thus, by forging ICMPv6 "Packet Too Big" error messages an attacker can cause hosts to employ "atomic fragments", and then launch any fragmentation-based attacks against such traffic. This document discusses the generation of the aforementioned "atomic fragments", the corresponding security implications, and formally updates RFC 2460 and RFC 5722 such that fragmentation-based attack vectors against traffic employing "atomic fragments" are completely eliminated.

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] allowed 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] forbid 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* be actually fragmented into several pieces, but just include a Fragment Header with both the "Fragment Offset" and the "M" bit set to 0.

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 2 describes the generation of IPv6 "atomic fragments", and how they can be remotely "triggered" by a remote attacker. Section 3 formally updates [RFC2460] and [RFC5722] such that the aforementioned attack vector is eliminated.

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. 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].

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 [RFC5927].

> 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.

> Upon receipt of one of the aforementioned ICMPv6 "Packet Too Big" error messages, the Destinations Cache 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.

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

## 3. Updating RFC 2460 and RFC 5722

This document updates [RFC2460] and [RFC5722] as follows:

A host that receives an IPv6 packet which includes a Fragment Header with the "Fragment Offset" equal to 0 and the "M" bit equal to 0 MUST process such 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}. That is, the Fragment Header of "atomic fragments" should be removed by the receiving host, and the resulting packet should be processed as a non-fragmented IPv6 datagram. Additionally, any fragments already queued with the same set {IPV6 Source Address, IPv6 Destination Address, Fragment Identification} should not be discarded upon receipt of the "colliding" IPv6 atomic fragment, since IPv6 atomic fragments do not really interfere with "normal" fragmented traffic.

## 4. 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.

## 5. Security Considerations

This document describes how an attacker can exploit ICMPv6 "Packet Too Big" error messages to cause further IPv6 packets to include a Fragment Header, such that he can perform any fragmentation-based attack against otherwise non-fragmented traffic. This document updates [RFC2460] and [RFC5722], such that the aforementioned attack vector is completely eliminated.

## 6. Acknowledgements

The author would like to thank (in alphabetical order) Tore Anderson, Ran Atkinson, Remi Despres, Timothy Hartrick, Steinar Haug, Philip Homburg, Simon Perreault, 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).

Fernando Gont would like to thank CPNI (http://www.cpni.gov.uk) for their continued support.

# 7. References

## 7.1. 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.

[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.

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

## 7.2. Informative References

[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.

[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, December 2011, <http://tools.ietf.org/html/draft-gont-6man-predictable-fragment-id>.

## 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.

| Operating System | Processes 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 |
| OpenBSD-current | Yes | Yes |
| Solaris 11 | Yes | Yes |
| Windows 7 Home Premium | Yes | No |

Table 1: Processing of IPv6 atomic fragments by different OSes

## Appendix B. Changes from previous versions of the document (to be removed by the RFC Editor before publication of this document as a RFC

### B.1. Changes from draft-gont-6man-ipv6-atomic-fragments-00

- Miscellaneous editorial changes

- Clarified Section 3.

- Incorporates a survey of support of IPv6 "atomic fragments" in different OSes in Appendix A.

- Draft resubmitted as draft-ietf.

## Author's Address

Fernando Gont
UK CPNI

Email: fgont@si6networks.com
URI: http://www.cpni.gov.uk