           Generation of IPv6 Atomic Fragments Considered Harmful

Abstract

   This document discusses the security implications of the generation
   of IPv6 atomic fragments and a number of interoperability issues
   associated with IPv6 atomic fragments. It concludes that the
   aforementioned functionality is undesirable and thus documents the
   motivation for removing this functionality from an upcoming revision
   of the core IPv6 protocol specification (RFC 2460).

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   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). Not all documents
   approved by the IESG are a candidate for any level of Internet
   Standard; see Section 2 of RFC 7841.

   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/rfc8021.

Copyright Notice

Table of Contents

1. Introduction

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

   A legacy IPv4/IPv6 translator implementing the Stateless IP/ICMP
   Translation Algorithm [RFC6145] may legitimately generate ICMPv6
   "Packet Too Big" (PTB) error messages [RFC4443] advertising an MTU
   smaller than 1280 (the minimum IPv6 MTU). Section 5 of [RFC2460]
   states that, upon receiving such an ICMPv6 error message, hosts are
   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; rather, they will be "atomic" fragments [RFC6946]
   (i.e., they will just include a Fragment Header with both the
   "Fragment Offset" and the "M" flag set to 0). [RFC6946] requires
   that these atomic fragments be essentially processed by the
   destination host(s) as non-fragmented traffic (since there are not

really any fragments to be reassembled). The goal of these atomic
fragments is simply to convey an appropriate 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 be leveraged for
performing a number of attacks against the corresponding IPv6 flows.
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, for example, having
IPv6/IPv4 translators generate an IPv4 Identification value
themselves), we conclude that this functionality is undesirable.

Section 2 briefly discusses the security implications of the
generation of IPv6 atomic fragments and describes a specific
Denial-of-Service (DoS) attack vector that leverages the widespread
dropping of IPv6 fragments in the public Internet. Section 3
provides additional considerations regarding the usefulness of
generating IPv6 atomic fragments.

2. Security Implications of the Generation of IPv6 Atomic Fragments

The security implications of IP fragmentation have been discussed at
length in [RFC6274] and [RFC7739]. An attacker can leverage the
generation of IPv6 atomic fragments to trigger the use of
fragmentation in an arbitrary IPv6 flow (in scenarios in which actual
fragmentation of packets is not needed) and can subsequently perform
any type of fragmentation-based attack against legacy IPv6 nodes that
do not implement [RFC6946]. That is, employing fragmentation where
not actually needed allows for fragmentation-based attack vectors to
be employed, unnecessarily.

We note that, unfortunately, even nodes that already implement
[RFC6946] can be subject to DoS attacks as a result of the generation
of IPv6 atomic fragments. Let us assume that Host A is communicating
with Host B and that, as a result of the widespread dropping of IPv6
packets that contain extension headers (including fragmentation)
[RFC7872], some intermediate node filters fragments between Host B
and Host A. If an attacker sends a forged ICMPv6 PTB error message
to Host 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 Host B starts sending IPv6 atomic fragments (in
response to the received ICMPv6 PTB error message), these packets
will be dropped, since we previously noted that IPv6 packets with
extension headers were being dropped between Host B and Host A.
Thus, this situation will result in a DoS scenario.

Another possible scenario is that in which two BGP peers are
employing IPv6 transport and they implement Access Control Lists
(ACLs) to drop IPv6 fragments (to avoid control-plane attacks). If
the aforementioned BGP peers drop IPv6 fragments but still honor
received ICMPv6 PTB error messages, an attacker could easily attack
the corresponding peering session by simply sending an ICMPv6 PTB
message with a reported MTU smaller than 1280 bytes. Once the attack
packet has been sent, the aforementioned routers will themselves be
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 filters as per BCP 38
   [BCP38] does not help as a countermeasure.

o Only the IPv6 addresses of the IPv6 packet embedded in the ICMPv6
   payload need to be forged. While one could envision filtering
   devices enforcing filters in the style of BCP 38 on the ICMPv6
   payload, the use of extension headers (by the attacker) could make
   this difficult, if not impossible.

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

o As noted in Section 3, SIIT (the Stateless IP/ICMP Translation
   Algorithm) [RFC6145], including derivative protocols such as
   Stateful NAT64 (Network Address and Protocol Translation from IPv6
   Clients to IPv4 Servers) [RFC6146], was the only technology making
   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] 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.

3. Additional Considerations

   Besides the security assessment provided in Section 2, 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 destination
       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

   4. The ability of the translator implementation to access the
       information conveyed by the Fragment Header

   5. The value extracted from the low-order 16 bits of the IPv6
       fragment header Identification field resulting in an appropriate
       IPv4 Identification value

   Unfortunately,

   1. There exists a fair share of evidence of ICMPv6 PTB error
       messages being dropped on the public Internet (for instance, that
       is one of the reasons for which Packetization Layer Path MTU
       Discovery (PLPMTUD) [RFC4821] was produced). Therefore, relying
       on such messages being successfully delivered will affect the
       robustness of the protocol that relies on them.

   2. 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. 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.

3. IPv6 atomic fragment generation represents a case in which
    fragmented traffic is produced where otherwise it would not be
    needed. Since there is widespread dropping of IPv6 fragments in
    the public Internet [RFC7872], this would mean that the
    (unnecessary) use of IPv6 fragmentation might result,
    unnecessarily, in a DoS situation even in legitimate cases.

4. The packet-handling API at the node where the translator is
    running may obscure fragmentation-related information. In such
    scenarios, the information conveyed by the Fragment Header may be
    unavailable to the translator. [JOOL] discusses a sample
    framework (Linux Netfilter) that hinders access to the
    information conveyed in IPv6 fragments.

5. While [RFC2460] requires that the IPv6 fragment header
    Identification field of a fragmented packet be different than
    that of any other fragmented packet sent recently with the same
    Source Address and Destination Address, there is no requirement
    on the low-order 16 bits of such a value. Thus, there is no
    guarantee that IPv4 fragment Identification collisions will be
    avoided or reduced by employing the low-order 16 bits of the IPv6
    fragment header Identification field of a packet sent by a source
    host. Besides, collisions might occur where two distinct IPv6
    Destination Addresses are translated into the same IPv4 address,
    such that Identification values that might have been generated to
    be unique in the context of IPv6 end up colliding when used in
    the context of translated IPv4.

We note that SIIT essentially employs the Fragment Header of IPv6
atomic fragments to signal the translator how to set the Don't
Fragment (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 a Fragment Header). Additionally,
the translator will employ the low-order 16 bits of the IPv6 fragment
header Identification field for setting the IPv4 Identification. At
least in theory, this is expected to reduce the IPv4 Identification
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 smaller
    than 1260 bytes. An IPv4 Path MTU of 1260 corresponds to an IPv6
    Path MTU of 1280, due to an optionless IPv4 header being 20 bytes
    shorter than the IPv6 header.

   3. ECMP routing [RFC2992] with more than one translator is employed,
       for example, for redundancy purposes.

   In such a scenario, if each translator were to select the IPv4
   Identification on its own (rather than selecting the IPv4
   Identification from the low-order 16 bits of the fragment
   Identification of IPv6 atomic fragments), this could possibly lead to
   IPv4 Identification collisions. However, as noted above, the value
   extracted from the low-order 16 bits of the IPv6 fragment header
   Identification field might not result in an appropriate IPv4
   Identification: for example, a number of implementations set the IPv6
   fragment header Identification field according to the output of a
   Pseudorandom Number Generator (PRNG) (see Appendix B of [RFC7739]);
   hence, if the translator only employs the low-order 16 bits of such a
   value, it is very unlikely that relying on the fragment
   Identification of the IPv6 atomic fragment will result in a reduced
   IPv4 Identification collision rate (when compared to the case where
   the translator selects each IPv4 Identification on its own).
   Besides, because of the limited size of the IPv4 Identification
   field, it is nevertheless virtually impossible to guarantee
   uniqueness of the IPv4 Identification values without artificially
   limiting the data rate of fragmented traffic [RFC6864] [RFC4963].

   [RFC6145] was the only "consumer" of IPv6 atomic fragments, and it
   correctly and diligently noted (in its Section 6) the possible
   interoperability problems of relying on IPv6 atomic fragments,
   proposing a workaround that led to more robust behavior and
   simplified code. [RFC6145] has been obsoleted by [RFC7915], such
   that SIIT does not rely on IPv6 atomic fragments.

4. Conclusions

   Taking all of the above considerations into account, we recommend
   that IPv6 atomic fragments be deprecated.

   In particular:

   o IPv4/IPv6 translators should be updated to not generate ICMPv6 PTB
      error messages containing an MTU value smaller than the minimum
      IPv6 MTU of 1280 bytes. This will ensure that current IPv6 nodes
      will never have a legitimate need to start generating IPv6 atomic
      fragments.

   o The recommendation in the previous bullet ensures that there are
      no longer any valid reasons for ICMPv6 PTB error messages
      reporting an MTU value smaller than the minimum IPv6 MTU
      (1280 bytes). IPv6 nodes should therefore be updated to ignore
      ICMPv6 PTB error messages reporting an MTU smaller than 1280 bytes
      as invalid.

   We note that these recommendations have been incorporated in
   [IPv6-PMTUD], [IPv6-Spec], and [RFC7915].

5. Security Considerations

   This document briefly discusses the security implications of the
   generation of IPv6 atomic fragments and describes one specific DoS
   attack vector that leverages the widespread dropping of IPv6
   fragments in the public Internet. It concludes that the generation
   of IPv6 atomic fragments is an undesirable feature and documents the
   motivation for removing this functionality from [IPv6-Spec].

6. References

6.1. Normative References

   [RFC2460]   Deering, S. and R. Hinden, "Internet Protocol, Version 6
               (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
               December 1998, <http://www.rfc-editor.org/info/rfc2460>.

   [BCP38]   Ferguson, P. and D. Senie, "Network Ingress Filtering:
               Defeating Denial of Service Attacks which employ IP Source
               Address Spoofing", BCP 38, RFC 2827, May 2000,
               <http://www.rfc-editor.org/info/rfc2827>.

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

   [RFC4821]   Mathis, M. and J. Heffner, "Packetization Layer Path MTU
               Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007,
               <http://www.rfc-editor.org/info/rfc4821>.

   [RFC4861]   Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
               "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
               DOI 10.17487/RFC4861, September 2007,
               <http://www.rfc-editor.org/info/rfc4861>.

   [RFC6145]   Li, X., Bao, C., and F. Baker, "IP/ICMP Translation
               Algorithm", RFC 6145, DOI 10.17487/RFC6145, April 2011,
               <http://www.rfc-editor.org/info/rfc6145>.

   [RFC7915]   Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont,
               "IP/ICMP Translation Algorithm", RFC 7915,
               DOI 10.17487/RFC7915, June 2016,
               <http://www.rfc-editor.org/info/rfc7915>.

   [RFC6864]   Touch, J., "Updated Specification of the IPv4 ID Field",
               RFC 6864, DOI 10.17487/RFC6864, February 2013,
               <http://www.rfc-editor.org/info/rfc6864>.

6.2. Informative References

   [RFC2992]   Hopps, C., "Analysis of an Equal-Cost Multi-Path
               Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000,
               <http://www.rfc-editor.org/info/rfc2992>.

   [RFC5927]   Gont, F., "ICMP Attacks against TCP", RFC 5927,
               DOI 10.17487/RFC5927, July 2010,
               <http://www.rfc-editor.org/info/rfc5927>.

   [RFC4963]   Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly
               Errors at High Data Rates", RFC 4963,
               DOI 10.17487/RFC4963, July 2007,
               <http://www.rfc-editor.org/info/rfc4963>.

   [RFC6052]   Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X.
               Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052,
               DOI 10.17487/RFC6052, October 2010,
               <http://www.rfc-editor.org/info/rfc6052>.

   [RFC6146]   Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
               NAT64: Network Address and Protocol Translation from IPv6
               Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146,
               April 2011, <http://www.rfc-editor.org/info/rfc6146>.

   [RFC6274]   Gont, F., "Security Assessment of the Internet Protocol
               Version 4", RFC 6274, DOI 10.17487/RFC6274, July 2011,
               <http://www.rfc-editor.org/info/rfc6274>.

   [RFC6946]   Gont, F., "Processing of IPv6 "Atomic" Fragments",
               RFC 6946, DOI 10.17487/RFC6946, May 2013,
               <http://www.rfc-editor.org/info/rfc6946>.

   [RFC7739]   Gont, F., "Security Implications of Predictable Fragment
               Identification Values", RFC 7739, DOI 10.17487/RFC7739,
               February 2016, <http://www.rfc-editor.org/info/rfc7739>.

   [RFC7872]   Gont, F., Linkova, J., Chown, T., and W. Liu,
               "Observations on the Dropping of Packets with IPv6
               Extension Headers in the Real World", RFC 7872,
               DOI 10.17487/RFC7872, June 2016,
               <http://www.rfc-editor.org/info/rfc7872>.

   [IPv6-Spec]
              Deering, S. and R. Hinden, "Internet Protocol, Version 6
              (IPv6) Specification", Work in Progress,
              draft-ietf-6man-rfc2460bis-08, November 2016.

   [IPv6-PMTUD]
              McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed.,
              "Path MTU Discovery for IP version 6", Work in Progress,
              draft-ietf-6man-rfc1981bis-03, October 2016.

   [JOOL]   Leiva Popper, A., "nf_defrag_ipv4 and nf_defrag_ipv6",
              April 2015, <https://github.com/NICMx/Jool/wiki/
              nf_defrag_ipv4-and-nf_defrag_ipv6#implementation-gotchas>.

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

   Will (Shucheng) Liu
   Huawei Technologies
   Bantian, Longgang District
   Shenzhen 518129
   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