                            The Null Packet

Abstract

   The ever-elusive Null Packet received numerous mentions in documents
   in the RFC series, but it has never been explicitly defined. This
   memo corrects that omission.

Status of This Memo

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

   This is a contribution to the RFC Series, independently of any other
   RFC stream. The RFC Editor has chosen to publish this document at
   its discretion and makes no statement about its value for
   implementation or deployment. Documents approved for publication by
   the RFC Editor are not a candidate for any level of Internet
   Standard; see 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/rfc6592.

Table of Contents

1. Introduction

   Null Packets are neither sent nor acknowledged when not received.
   They are perfect in their simplicity and they are very true, as they
   extrapolate from the twelfth Truth of networking [RFC1925]: there is
   literally nothing left to take away.

   An early mention of the Null Packet is attributed to Van Jacobson in
   the context of TCP/IP Header Compression [RFC1144]. Mind you, the
   Null Packet is not created by compressing a packet until it
   disappears into nothingness. Such a compression scheme might not be
   reversible; instead, Section 3.2.4 of [RFC1144] describes an explicit
   lack of response as "Nothing (a null packet) is returned".

   Many documents attempt to define in-the-wire code points and protocol
   identifiers (PIDs) for a Null Packet [RFC4259] [RFC4571] [RFC5320].
   However, such an exercise is futile. This memo postulates that a
   Null Packet cannot have a PID, as the existence of a protocol
   construct or value would null the null; this includes the inability
   to use 0x0, 0x0000, or even 0x00000000, but excludes the restriction
   to use "" (see Section 2.1).

   An IPv6 Next Header value of 59 (No Next Header) (see Section 4.7 of
   [RFC2460]) does not create a Null Packet.

2. The Null Packet

   The Null Packet is a zero-dimensional packet. The Null Packet exists
   since it is non-self-contradictorily definable.

2.1. Formal Definition

   [This section is intentionally left blank, see also Section 0 of
   [NULL].]

2.2. Faux Amis

   Many experts naively confuse the Null Packet with an Imaginary
   Packet, in a rationalization attempt when faced with the inability to
   prove the existence of the Null Packet. For reference, an Imaginary
   Packet contains the IP Version of 4i or 6i. However, protocol
   purists are not fooled and quickly plea with experts to get real.

   The Null Packet's qualities should not be confused with the bit-
   bucket blackhole nature of the null device, since the Null Packet
   does not discard packets. Confusion might stem from the fact that
   the behavior is similar to that of input streams reading from /dev/
   null (i.e., "nothing is returned").

3. Performance Metrics Considerations

   A protocol sending Null Packets effectively sends packets of zero
   length. One characteristic of flow streams of Null Packet traffic is
   that increasing the rate at which Null Packets are sent does not
   increase the bit rate of the Null Packet traffic. The bit rate
   continues being unequivocally null, unless an infinite number of Null
   Packets per unit of time could be sent. Similarly, should a user
   stop sending Null Packets, the bit rate of Null Packets would not
   vary. Traditional traffic performance metrics are not well suited to
   qualify Null Packet traffic; this fact argues for the creation of new
   sets of performance metrics that test positive for "usefulness" (see
   Section 5.2 of [RFC6390]).

4. Security Considerations

   When used in a Multiprotocol Label Switching (MPLS) environment, the
   Null Packet can only use an Implicit NULL label (see Section 4.1.5 of
   [RFC3031]. The Implicit NULL label is a label that can be
   distributed, but which never actually appears in the encapsulation.
   The Nil FEC is not used.

   The security considerations for the Null Packet are undefined, as
   hereby described. The "good" nature of Null Packets is quite
   useless, and the "bad" nature of Null Packets is rather inefficient.

4.1. The Paradoxical Firewall

   Many firewalls and other security devices have trouble identifying
   the Null Packet. Others claim to filter out Null Packets quite
   effectively and effortlessly. Interestingly, or not, both might be
   correct, which begs the omnipotence paradox: Can a firewall create a
   rule to filter out the Null Packet coming from the "outside", and not
   see Null Packets being allowed on the "inside"?

4.2. The Null Packet is Good

   The Null Packet cannot have the Evil Bit ("E") [RFC3514] set, by
   definition (see Section 2.1). Consequently, it is rather clear and
   undeniable that the Null Packet is harmless, having no evil intent.

4.3. Just Encrypt It, Carefully

   A commonly accepted practice for Security Considerations sections is
   to wrap a blanket "encrypt around foo" statement, for almost any
   value of "foo". This document is no exception. However, surgical
   care must be taken to not apply NULL encryption [RFC2410] to the Null
   Packet; such a careless act can bring discontinuities and "Oops" more
   epic than dividing by zero or Googling the word "Google" (it has been
   rumored that such action can break the Internet, although this can be
   easily disproved by reductio ad absurdum.)

4.4. Denial of Denial of Service

   Even when sysadmins, netadmins, secadmins, and other NOC engineers
   are faced with the undisputed inability to block Null Packets (see
   Section 4.1), attacks leveraging Null Packets are not quite so common
   in the wild and are not seen in the seek^Wsecurity news. Perhaps
   because these unusual packets are hard to spoof in the data plane, or
   because their Time to Live (TTL) or Hop Limit cannot be altered since
   it does not exist [RFC5082], the fact is that Null Packets present a
   denial of denial of service (DoDoS).

   An important corollary is that dropping Null Packets does not
   generate packets.

5. IANA Considerations

   This document explicitly and emphatically, yet very humbly, requests
   IANA to not create an empty registry for the Null Packet.

6. References

6.1. Normative References

   [NULL]   "".

   [RFC1144] Jacobson, V., "Compressing TCP/IP headers for low-speed
              serial links", RFC 1144, February 1990.

   [RFC1925] Callon, R., "The Twelve Networking Truths", RFC 1925,
              April 1996.

   [RFC3514] Bellovin, S., "The Security Flag in the IPv4 Header",
              RFC 3514, April 1 2003.

6.2. Informative References

   [RFC2410] Glenn, R. and S. Kent, "The NULL Encryption Algorithm and
              Its Use With IPsec", RFC 2410, November 1998.

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

   [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol
              Label Switching Architecture", RFC 3031, January 2001.

   [RFC4259] Montpetit, M., Fairhurst, G., Clausen, H., Collini-Nocker,
              B., and H. Linder, "A Framework for Transmission of IP
              Datagrams over MPEG-2 Networks", RFC 4259, November 2005.

   [RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP)
              and RTP Control Protocol (RTCP) Packets over Connection-
              Oriented Transport", RFC 4571, July 2006.

   [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., and C.
              Pignataro, "The Generalized TTL Security Mechanism
              (GTSM)", RFC 5082, October 2007.

   [RFC5320] Templin, F., "The Subnetwork Encapsulation and Adaptation
              Layer (SEAL)", RFC 5320, February 2010.

   [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New
              Performance Metric Development", BCP 170, RFC 6390,
              October 2011.

Author's Address

     Carlos Pignataro
     Cisco Systems, Inc.
     7200-12 Kit Creek Road
     Research Triangle Park, NC   27709
     US

     EMail: cpignata@cisco.com