Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on December 14, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

The Hop-by-Hop option header is a type of IPv6 extension header that has been defined in the IPv6 protocol specification. The contents of this header need to be processed by every node along the path of an IPv6 datagram. This draft highlights the characteristics of this extension header which make it prone to Denial of Service attacks and proposes an arrangement of options to minimize such attacks.
Table of Contents

1. Introduction ................................................... 3
   1.1 Conventions used in this document ........................ 3
2. Details of the attack .............................................. 4
   2.1 Effects of the attack ...................................... 4
3. Optimal arrangement of options ................................. 5
4. Proposed arrangement of options ................................. 6
5. Deployment Considerations ....................................... 7
   5.1 Impact on deployed IPv6 nodes ............................ 7
   5.2 Alternate solutions ....................................... 7
   5.3 Quantitative analysis ..................................... 7
6. Security Considerations ......................................... 8
7. References ...................................................... 8
   Author’s Address ............................................. 8
   Intellectual Property and Copyright Statements ............... 9
1. Introduction

The IPv6 base specification [RFC2460] defines the hop-by-hop extension header. This extension header carries the options which need to be processed by every node along the path of the datagram. Certain characteristics of the specification make it especially vulnerable to Denial of Service attacks. The characteristics are:

- All the ipv6 nodes on the path need to process the options in this header.
- The option TLVs in the hop-by-hop options header need to be processed in order.
- A sub range of option types in this header will not cause any errors even if the node does not recognize them.
- There is no restriction as to how many occurrences of an option type can be present in the hop-by-hop header.

This document details a low bandwidth Denial of Service attack on ipv6 routers/hosts using the hop-by-hop options extension header and possible ways of mitigating these attacks.

1.1 Conventions used in this document

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 [RFC2119].
2. Details of the attack

The denial of service attack can be carried out by forming an IP datagram with a large number of TLV encoded options with random option type identifiers in the hop-by-hop options header. The option type is a 8 bit field with special meaning attached to the three most significant bits. The attack is most effective when all the nodes in the path are affected, meaning we do not want any node to drop the packet and send ICMP errors regarding unrecognized options. If the two most significant bits are cleared(0), the receiving node will silently ignore the option if it does not recognize the option type. The third most significant bit is used to denote whether the option data can change en-route. If the bit is set to 1 the option data can change en route. The attack is equally effective whether or not an IPSec Authentication Header(AH) treats the option data as zero valued octets. Hence we can include this bit in generating option types. The acceptable option types would be laid out like below

```
/+-----------------+-----------------+-----------------+-----------------+
| Option Type     | Opt Data Len    | Option Data     |
| 0 0 x x x x x x|x.............|.................|
+-----------------+-----------------+-----------------+
```

Figure 1: Option type layout

Since the option types 0(0x00) and 1(0x01) are reserved for the Pad1 and PadN options in [RFC2460] we exclude these from the acceptable range as well. So we choose the option type identifiers for each of these options to be in the range 0x02-0x63. More option types defined by other RFCs can be excluded from the attack as and when they are allocated by the IANA. Examples are Tunnel Encapsulation limit (0x04) and Router Alert (0x05).

2.1 Effects of the attack

The attack can be used to cripple the routers by attacking the control processor rather than the forwarding plane. Since the control traffic, like the routing protocols, shares the same resources with this traffic, this kind of attack may be hard to control. On routers having separate Control and Forwarding elements only the Control traffic would be affected. For routers whose the Control and Forwarding elements are fused together this would lead to problems with forwarding packets as well.
3. Optimal arrangement of options

This attack can be mitigated by restricting each option type to occur only once in a given extension header. Since it would be computationally expensive for each IPv6 node to remember all the option types which have already occurred in the header, it makes sense to specify some kind of ordering of the option type identifiers within the hop by hop options header. The most efficient arrangement is to arrange the options in descending order of the numerical value of the option type identifier. With this arrangement it is trivial to check if an option has occurred before. The IPv6 node has to remember and compare to only the last encountered option in addition to the current one rather than remembering and comparing all the previously encountered options. The reserved option types 0x00 (Pad1) and 0x01 (PadN) options MAY occur anywhere in the header and they are not considered for this check.

This exception leaves the door open for another class of DoS attacks. The attacker can use just the Pad1 and PadN options multiple times in the header to achieve the same effect as the previously detailed attack. Hence it becomes necessary to add one more restriction. Two padding options MUST NOT appear together in the header. Since there is no legitimate case where two pad options have to appear together this will not cause too many problems.

The option type identifier space is shared between the hop-by-hop options and the destination options extension headers. Therefore the attack is equally applicable to the destination options header but is not as effective because only the destination node processes the header. Similar language MAY be used to specify the destination options header as well.
4. Proposed arrangement of options

Within an IPv6 hop-by-hop option header each option type MUST NOT occur more than once with the exception of the Pad1(0x00) and PadN(0x01) options. The options MUST be arranged in descending order of the numerical value of the option type identifier with the exception of the Pad1 and the PadN options. The pad options MAY occur anywhere in the header but two pad options MUST be separated by an option which is not a pad option.

So the final receiving algorithm looks like this

```c
/* Receiving algorithm for hop-by-hop options */
last_option_was_pad=0;
first_option=1;
while (more_options) {
    if (current_option_id & 0xfe) {
        if (last_option_was_pad) {
            /* Error: Cannot have two pad options together */
            /* Send ICMPv6 message */
        } else {
            last_option_was_pad=1;
            continue;
        }
    }
    if (current_option_id<previous_option_id) {
        previous_option_id=current_option_id;
        last_option_was_pad=0;
        /* Process the option */
    } else {
        if (first_option) {
            first_option=0;
            previous_option_id=current_option_id;
            last_option_was_pad=0;
            /* Process the option */
        }
        /* Error option is duplicate or out of order*/
        /* Send ICMPv6 message */
    }
}
```

Figure 2: Receiving algorithm
5. Deployment Considerations

5.1 Impact on deployed IPv6 nodes

The proposed changes affect all currently IPv6 nodes which need to send packets with hop-by-hop options. The IPv6 stack on these nodes needs to be modified to send out the options in the correct order. Since there are not too many option types which are currently supported by deployed nodes, it is very likely that the nodes need to be updated anyway for supporting more option types as they are assigned.

5.2 Alternate solutions

There are other possible solutions to handle the DoS attack mentioned in this draft. The first one that comes to mind is to simply rate limit packets with hop-by-hop option headers and start dropping them randomly when the CPU load becomes very high. While this solution is very simple and has no impact on deployed IPv6 nodes, it is sub-optimal. A legitimate packet with a hop-by-hop option header has the same probability of being dropped as an attack packet. Implementing the solution proposed in this draft does not preclude the use of rate limiting. In fact it gives a legitimate packet a lower probability of being dropped, since most of the obvious attack traffic would have been dropped by the receiving algorithm.

5.3 Quantitative analysis

The proposed solution gives cuts processing times in worst case scenarios by between 2x-8x depending on how many options in the 0x64-0xff range are allocated by the IANA. The reduction in processing times is inversely proportional to the number of options allocated in this range. The 2x number is valid when all the 192 options have been allocated and the 8x number applies when none of them are allocated.
6. Security Considerations

This document highlights the possible security issues with the IPv6 hop-by-hop option header specified in [RFC2460] which can lead to denial of service attacks and suggests some changes to reduce the effect of the DoS attacks.

7 References


Author’s Address

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

EMail: suresh.krishnan@ericsson.com
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the
Internet Society.