Network Working Group
Request for Comments: 3514
Category: Informational

S. Bellovin
AT&T Labs Research

1 April 2003

The Security Flag in the IPv4 Header

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

Firewalls, packet filters, intrusion detection systems, and the like often have difficulty distinguishing between packets that have malicious intent and those that are merely unusual. We define a security flag in the IPv4 header as a means of distinguishing the two cases.

1. Introduction

Firewalls [CBR03], packet filters, intrusion detection systems, and the like often have difficulty distinguishing between packets that have malicious intent and those that are merely unusual. The problem is that making such determinations is hard. To solve this problem, we define a security flag, known as the "evil" bit, in the IPv4 [RFC791] header. Benign packets have this bit set to 0; those that are used for an attack will have the bit set to 1.

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. Syntax

The high-order bit of the IP fragment offset field is the only unused bit in the IP header. Accordingly, the selection of the bit position is not left to IANA.

The bit field is laid out as follows:

Currently-assigned values are defined as follows:

0x0 If the bit is set to 0, the packet has no evil intent. Hosts, network elements, etc., SHOULD assume that the packet is harmless, and SHOULD NOT take any defensive measures. (We note that this part of the spec is already implemented by many common desktop operating systems.)

0x1 If the bit is set to 1, the packet has evil intent. Secure systems SHOULD try to defend themselves against such packets. Insecure systems MAY chose to crash, be penetrated, etc.

3. Setting the Evil Bit

There are a number of ways in which the evil bit may be set. Attack applications may use a suitable API to request that it be set. Systems that do not have other mechanisms MUST provide such an API; attack programs MUST use it.

Multi-level insecure operating systems may have special levels for attack programs; the evil bit MUST be set by default on packets emanating from programs running at such levels. However, the system MAY provide an API to allow it to be cleared for non-malicious activity by users who normally engage in attack behavior.

Fragments that by themselves are dangerous MUST have the evil bit set. If a packet with the evil bit set is fragmented by an intermediate router and the fragments themselves are not dangerous, the evil bit MUST be cleared in the fragments, and MUST be turned back on in the reassembled packet.

Intermediate systems are sometimes used to launder attack connections. Packets to such systems that are intended to be relayed to a target SHOULD have the evil bit set.

Some applications hand-craft their own packets. If these packets are part of an attack, the application MUST set the evil bit by itself.

In networks protected by firewalls, it is axiomatic that all attackers are on the outside of the firewall. Therefore, hosts inside the firewall MUST NOT set the evil bit on any packets.

Because NAT [RFC3022] boxes modify packets, they SHOULD set the evil bit on such packets. "Transparent" http and email proxies SHOULD set the evil bit on their reply packets to the innocent client host.

Some hosts scan other hosts in a fashion that can alert intrusion detection systems. If the scanning is part of a benign research project, the evil bit MUST NOT be set. If the scanning per se is innocent, but the ultimate intent is evil and the destination site has such an intrusion detection system, the evil bit SHOULD be set.

4. Processing of the Evil Bit

Devices such as firewalls MUST drop all inbound packets that have the evil bit set. Packets with the evil bit off MUST NOT be dropped. Dropped packets SHOULD be noted in the appropriate MIB variable.

Intrusion detection systems (IDSs) have a harder problem. Because of their known propensity for false negatives and false positives, IDSs MUST apply a probabilistic correction factor when evaluating the evil bit. If the evil bit is set, a suitable random number generator [RFC1750] must be consulted to determine if the attempt should be logged. Similarly, if the bit is off, another random number generator must be consulted to determine if it should be logged despite the setting.

The default probabilities for these tests depends on the type of IDS. Thus, a signature-based IDS would have a low false positive value but a high false negative value. A suitable administrative interface MUST be provided to permit operators to reset these values.

Routers that are not intended as as security devices SHOULD NOT examine this bit. This will allow them to pass packets at higher speeds.

As outlined earlier, host processing of evil packets is operating-system dependent; however, all hosts MUST react appropriately according to their nature.

5. Related Work

Although this document only defines the IPv4 evil bit, there are complementary mechanisms for other forms of evil. We sketch some of those here.

For IPv6 [RFC2460], evilness is conveyed by two options. The first, a hop-by-hop option, is used for packets that damage the network, such as DDoS packets. The second, an end-to-end option, is for packets intended to damage destination hosts. In either case, the

option contains a 128-bit strength indicator, which says how evil the packet is, and a 128-bit type code that describes the particular type of attack intended.

Some link layers, notably those based on optical switching, may bypass routers (and hence firewalls) entirely. Accordingly, some link-layer scheme MUST be used to denote evil. This may involve evil lambdas, evil polarizations, etc.

DDoS attack packets are denoted by a special diffserv code point.

An application/evil MIME type is defined for Web- or email-carried mischief. Other MIME types can be embedded inside of evil sections; this permit easy encoding of word processing documents with macro viruses, etc.

6. IANA Considerations

This document defines the behavior of security elements for the 0x0 and 0x1 values of this bit. Behavior for other values of the bit may be defined only by IETF consensus [RFC2434].

7. Security Considerations

Correct functioning of security mechanisms depend critically on the evil bit being set properly. If faulty components do not set the evil bit to 1 when appropriate, firewalls will not be able to do their jobs properly. Similarly, if the bit is set to 1 when it shouldn't be, a denial of service condition may occur.

8. References

[CBR03] W.R. Cheswick, S.M. Bellovin, and A.D. Rubin, "Firewalls and Internet Security: Repelling the Wily Hacker", Second Edition, Addison-Wesley, 2003.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1750] Eastlake, D., 3rd, Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

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

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

9. Author's Address

Steven M. Bellovin
AT&T Labs Research
Shannon Laboratory
180 Park Avenue
Florham Park, NJ 07932

Phone: +1 973-360-8656
EMail: bellovin@acm.org

10. Full Copyright Statement

Acknowledgement

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