Network Working Group
Request for Comments: 2873
Category: Standards Track

X. Xiao
Global Crossing
A. Hannan
iVMG
V. Paxson
ACIRI/ICSI
E. Crabbe
Exodus Communications

June 2000

TCP Processing of the IPv4 Precedence Field

Status of this Memo

Copyright Notice

Abstract

This memo describes a conflict between TCP [RFC793] and DiffServ [RFC2475] on the use of the three leftmost bits in the TOS octet of an IPv4 header [RFC791]. In a network that contains DiffServ-capable nodes, such a conflict can cause failures in establishing TCP connections or can cause some established TCP connections to be reset undesirably. This memo proposes a modification to TCP for resolving the conflict.

Because the IPv6 [RFC2460] traffic class octet does not have any defined meaning except what is defined in RFC 2474, and in particular does not define precedence or security parameter bits, there is no conflict between TCP and DiffServ on the use of any bits in the IPv6 traffic class octet.

1. Introduction

In TCP, each connection has a set of states associated with it. Such states are reflected by a set of variables stored in the TCP Control Block (TCB) of both ends. Such variables may include the local and remote socket number, precedence of the connection, security level and compartment, etc. Both ends must agree on the setting of the precedence and security parameters in order to establish a connection and keep it open.

There is no field in the TCP header that indicates the precedence of a segment. Instead, the precedence field in the header of the IP packet is used as the indication. The security level and compartment are likewise carried in the IP header, but as IP options rather than a fixed header field. Because of this difference, the problem with precedence discussed in this memo does not apply to them.

TCP requires that the precedence (and security parameters) of a connection must remain unchanged during the lifetime of the connection. Therefore, for an established TCP connection with precedence, the receipt of a segment with different precedence indicates an error. The connection must be reset [RFC793, pp. 36, 37, 40, 66, 67, 71].

With the advent of DiffServ, intermediate nodes may modify the Differentiated Services Codepoint (DSCP) [RFC2474] of the IP header to indicate the desired Per-hop Behavior (PHB) [RFC2475, RFC2597, RFC2598]. The DSCP includes the three bits formerly known as the precedence field. Because any modification to those three bits will be considered illegal by endpoints that are precedence-aware, they may cause failures in establishing connections, or may cause established connections to be reset.

2. Terminology

Segment: the unit of data that TCP sends to IP

Precedence Field: the three leftmost bits in the TOS octet of an IPv4 header. Note that in DiffServ, these three bits may or may not be used to denote the precedence of the IP packet. There is no precedence field in the traffic class octet in IPv6.

TOS Field: bits 3-6 in the TOS octet of IPv4 header [RFC 1349].

MBZ field: Must Be Zero

The structure of the TOS octet is depicted below:

```
                 0     1     2     3     4     5     6     7
              +-----+-----+-----+-----+-----+-----+-----+-----+
              |   PRECEDENCE    |          TOS          | MBZ |
              +-----+-----+-----+-----+-----+-----+-----+-----+
```

DS Field: the TOS octet of an IPv4 header is renamed the Differentiated Services (DS) Field by DiffServ.

The structure of the DS field is depicted below:

```
      0   1   2   3   4   5   6   7
    +---+---+---+---+---+---+---+---+
    |         DSCP          |  CU   |
    +---+---+---+---+---+---+---+---+
```

DSCP: Differentiated Service Code Point, the leftmost 6 bits in the DS field.

CU: currently unused.

Per-hop Behavior (PHB): a description of the externally observable forwarding treatment applied at a differentiated services-compliant node to a behavior aggregate.

3. Problem Description

The manipulation of the DSCP to achieve the desired PHB by DiffServ-capable nodes may conflict with TCP’s use of the precedence field. This conflict can potentially cause problems for TCP implementations that conform to RFC 793. First, page 36 of RFC 793 states:

> If the connection is in any non-synchronized state (LISTEN, SYN-SENT, SYN-RECEIVED), and the incoming segment acknowledges something not yet sent (the segment carries an unacceptable ACK), or if an incoming segment has a security level or compartment which does not exactly match the level and compartment requested for the connection, a reset is sent. If our SYN has not been acknowledged and the precedence level of the incoming segment is higher than the precedence level requested then either raise the local precedence level (if allowed by the user and the system) or send a reset; or if the precedence level of the incoming segment is lower than the precedence level requested then continue as if the precedence matched exactly (if the remote TCP cannot raise the precedence level to match ours this will be detected in the next segment it sends, and the connection will be terminated then). If our SYN has been acknowledged (perhaps in this incoming segment) the precedence level of the incoming segment must match the local precedence level exactly, if it does not a reset must be sent.

This leads to Problem #1: For a precedence-aware TCP module, if during TCP’s synchronization process, the precedence fields of the SYN and/or ACK packets are modified by the intermediate nodes, resulting in the received ACK packet having a different precedence from the precedence picked by this TCP module, the TCP connection cannot be established, even if both modules actually agree on an identical precedence for the connection.

Then, on page 37, RFC 793 states:

> If the connection is in a synchronized state (ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME-WAIT), security level, or compartment, or precedence which does not exactly match the level, and compartment, and precedence requested for the connection, a reset is sent and connection goes to the CLOSED state.

This leads to Problem #2: For a precedence-aware TCP module, if the precedence field of a received segment from an established TCP connection has been changed en route by the intermediate nodes so as to be different from the precedence specified during the connection setup, the TCP connection will be reset.

Each of problems #1 and #2 has a mirroring problem. They cause TCP connections that must be reset according to RFC 793 not to be reset.

Problem #3: A TCP connection may be established between two TCP modules that pick different precedence, because the precedence fields of the SYN and ACK packets are modified by intermediate nodes, resulting in both modules thinking that they are in agreement for the precedence of the connection.

Problem #4: A TCP connection has been established normally by two TCP modules that pick the same precedence. But in the middle of the data transmission, one of the TCP modules changes the precedence of its segments. According to RFC 793, the TCP connection must be reset. In a DiffServ-capable environment, if the precedence of the segments is altered by intermediate nodes such that it retains the expected value when arriving at the other TCP module, the connection will not be reset.

4. Proposed Modification to TCP

The proposed modification to TCP is that TCP must ignore the precedence of all received segments. More specifically:

(1) In TCP’s synchronization process, the TCP modules at both ends must ignore the precedence fields of the SYN and SYN ACK packets. The TCP connection will be established if all the conditions specified by RFC 793 are satisfied except the precedence of the connection.

(2) After a connection is established, each end sends segments with its desired precedence. The precedence picked by one end of the TCP connection may be the same or may be different from the precedence picked by the other end (because precedence is ignored during connection setup time). The precedence fields may be changed by the intermediate nodes too. In either case, the precedence of the received packets will be ignored by the other end. The TCP connection will not be reset in either case.

Problems #1 and #2 are solved by this proposed modification. Problems #3 and #4 become non-issues because TCP must ignore the precedence. In a DiffServ-capable environment, the two cases described in problems #3 and #4 should be allowed.

5. Security Considerations

A TCP implementation that terminates a connection upon receipt of any segment with an incorrect precedence field, regardless of the correctness of the sequence numbers in the segment's header, poses a serious denial-of-service threat, as all an attacker must do to terminate a connection is guess the port numbers and then send two segments with different precedence values; one of them is certain to terminate the connection. Accordingly, the change to TCP processing proposed in this memo would yield a significant gain in terms of that TCP implementation's resilience.

On the other hand, the stricter processing rules of RFC 793 in principle make TCP spoofing attacks more difficult, as the attacker must not only guess the victim TCP's initial sequence number, but also its precedence setting.

Finally, the security issues of each PHB group are addressed in the PHB group's specification [RFC2597, RFC2598].

6. Acknowledgments

Our thanks to Al Smith for his careful review and comments.

7. References

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

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1349] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.

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

[RFC2474] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC2597] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2587, June 1999.

[RFC2598] Jacobson, V., Nichols, K. and K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999.

8. Authors' Addresses

Xipeng Xiao
Global Crossing
141 Caspian Court
Sunnyvale, CA 94089
USA

Phone: +1 408-543-4801
EMail: xipeng@gblx.net

Alan Hannan
iVMG, Inc.
112 Falkirk Court
Sunnyvale, CA 94087
USA

Phone: +1 408-749-7084
EMail: alan@ivmg.net

Edward Crabbe
Exodus Communications
2650 San Tomas Expressway
Santa Clara, CA 95051
USA

Phone: +1 408-346-1544
EMail: edc@explosive.net

Vern Paxson
ACIRI/ICSI
1947 Center Street
Suite 600
Berkeley, CA 94704-1198
USA

Phone: +1 510-666-2882
EMail: vern@aciri.org

9. Full Copyright Statement

Acknowledgement

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