TCP Maintenance and Minor
Extensions Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2008

M. Bashyam
Ocarina Networks, Inc
M. Jethanandani
A. Ramaiah
Cisco Systems
June 28, 2008

Clarification of sender behaviour in persist condition.
draft-ananth-tcpm-persist-00.txt

Status of this Memo

Abstract

This document attempts to clarify the notion of the Zero Window Probes (ZWP) described in RFC 1122 [RFC1122]. In particular, it clarifies the actions that can be taken on connections which are experiencing the ZWP condition. The motivation for this document stems from the belief that TCP implementations strictly adhering to the current RFC language have the potential to become vulnerable to Denial of Service (DoS) scenarios.

Table of Contents

1. Introduction

RFC 1122 [RFC1122] Section 4.2.2.17, page 92 says that: A TCP MAY keep it's offered receive window closed indefinitely. As long as the receiving TCP continues to send acknowledgments in response to the probe segments, the sending TCP MUST allow the connection to stay open. The RFC goes on to say that it is important to remember that ACK (acknowledgement) segments that contain no data are not reliably transmitted by TCP. Therefore zero window probing SHOULD be supported to prevent a connection from hanging forever if ACK segments that re-opens the window is lost. The condition where the sender goes into the ZWP mode is typically known as the persist condition. The problem is applicable to TCP and TCP derived transport protocols like SCTP.

A consequence of adhering to the above requirement mandated by RFC 1122 is that multiple TCP receivers (clients) advertising a zero window to a busy server indefinitely (by reliably acknowledging the ZWP), could exhaust the connection and buffer resources of the sender (server). In such cases, to achieve robustness, the system should be able to take appropriate action on those TCP connections and reclaim resources. The purpose of this document is to clarify that such actions are in the spirit of RFC 1122 and they don't violate RFC 1122. The remainder of the document briefly describes the DoS scenario, analyzes the current verbiage surrounding the ZWP in RFC 1122 and attempts to disambiguate the notion presented by RFC 1122.

2. Discussion

Having the sender accumulate buffers and connection table entries when the receiver has deliberately and maliciously closed the window ultimately leads to resource exhaustion on the sender. This particular dependence on the receiver to open its zero window can be easily exploited by a malicious receiver TCP application to launch a DoS attack against the sender. In this scenario the sender's legitimate connections do not get established and already established well behaved TCP connections are unable to transmit any data. The sender enters the persist condition and is stuck waiting indefinitely for the receiver to open up its window.

To illustrate this, consider the case where the client application opens a TCP connection with a HTTP [RFC2616] server, sends a GET request for a large page and stops reading the response. This would cause the client TCP to advertise a zero window to the server. For every large HTTP response, the server is left holding on to all the response data in it's send queue. If the client never clears the persist condition, the server will continue to hold that data indefinitely. Multiple such TCP connections stuck in the same scenario on the server would cause resource depletion resulting in a DoS situation on the server.

In such scenarios it should be possible for the application or the system or a resource management entity to instruct TCP to terminate connections stalled in the persist condition. These actions are necessary to prevent resource exhaustion on the server.

An extensive discussion took place recently about this issue on the TCPM WG mailing list [TCPM]. The general opinion seemed to be that terminating a TCP connection in persist condition does not violate RFC 1122. In particular the operating system, a resource manager, or an application can instruct TCP to abort a connection in the persist condition. TCP itself SHOULD not take any action and continue to keep the connection open as mandated by RFC 1122 unless otherwise instructed to do so. The exact mechanism by which the instruction to abort the connection is conveyed to TCP is an implementation decision and falls beyond the scope of the current memo. To determine which TCP connection to abort the entity can use the connection attributes obtained from some interface similar to STATUS call mentioned in RFC 793.

3. Security Considerations

This memo primarily focuses on robustness of the system in general and robustness of TCP implementations during the persist condition in particular. This memo is intended to clarify that, actions like aborting TCP connections is well within the scope of the RFC 1122 language.

4. Conclusions

The document addresses the fact that terminating TCP connections stuck in the persist condition does not violate any RFC. It also suggests that TCP MUST not abort any connection until and unless explicitly requested to do so. The implementation details of the request is left to the implementer.

5. Acknowledgments

This document was inspired by the recent discussions that took place regarding the TCP persist condition issue in the TCPM WG mailing list [TCPM]. The outcome of those discussions was to come up with a draft that would clarify the intentions of the ZWP referred by RFC 1122. We would like to thank Mark Allman and David Borman for clarifying the objective behind this draft.

6. Informative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[TCPM] TCPM, "IETF TCPM Working Group and mailing list http://www.ietf.org/html.charters/tcpm-charter.html".

Authors' Addresses

Murali Bashyam
Ocarina Networks, Inc
42 Airport parkway
San Jose, CA 95110
USA

Phone: +1 (408) 512-2966
Email: mbashyam@ocarinanetworks.com

Mahesh Jethanandani
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Phone: +1 (408) 527-8230
Email: mahesh@cisco.com

Anantha Ramaiah
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Phone: +1 (408) 525-6486
Email: ananth@cisco.com

Full Copyright Statement

Intellectual Property