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

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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 30, 2008.

Abstract

This document attempts to clarify the notion of the Zero Window Probes (ZWP) described in <u>RFC 1122</u> [<u>RFC1122</u>]. 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

| <u>1</u> . | Introduction |
|------------|--|
| <u>2</u> . | Discussion |
| <u>3</u> . | Security Considerations |
| <u>4</u> . | Conclusions |
| <u>5</u> . | Acknowledgments |
| <u>6</u> . | Informative References |
| Auth | nors' Addresses |
| Inte | ellectual Property and Copyright Statements 10 |

Bashyam, et al. Expires December 30, 2008 [Page 2]

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.

Bashyam, et al. Expires December 30, 2008 [Page 3]

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.

[Page 4]

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 $\frac{\sf RFC\ 1122}{\sf I}$ 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 <u>RFC 1122</u>. We would like to thank Mark Allman and David Borman for clarifying the objective behind this draft.

TCP persist anomaly

<u>6</u>. Informative References

- [RFC1122] Braden, R., "Requirements for Internet Hosts -Communication Layers", STD 3, <u>RFC 1122</u>, October 1989.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>BCP 14</u>, <u>RFC 2119</u>, 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", <u>RFC 2616</u>, June 1999.
- [TCPM] TCPM, "IETF TCPM Working Group and mailing list http://www.ietf.org/html.charters/tcpm-charter.html".

Bashyam, et al. Expires December 30, 2008 [Page 8]

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

Bashyam, et al. Expires December 30, 2008 [Page 9]

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in $\frac{BCP}{78}$, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in <u>BCP 78</u> and <u>BCP 79</u>.

Copies of IPR disclosures made to the IETF Secretariat 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 implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.