

TCP Maintenance and Minor  
Extensions Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 11, 2010

M. Bashyam  
Ocarina Networks, Inc  
M. Jethanandani  
A. Ramaiah  
Cisco Systems  
July 10, 2009

Clarification of sender behaviour in persist condition.  
draft-ananth-tcpm-persist-01.txt

#### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [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/lid-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 January 11, 2010.

#### Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Internet-Draft

TCP persist anomaly

July 2009

## 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Discussion on <a href="#">RFC 1122</a> Requirement . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Description of Attack . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Clarification Regarding <a href="#">RFC 1122</a> Requirements . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Conclusion . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Acknowledgments . . . . .	<a href="#">8</a>
<a href="#">7.</a>	Programming Considerations . . . . .	<a href="#">9</a>
<a href="#">8.</a>	Informative References . . . . .	<a href="#">10</a>
	Authors' Addresses . . . . .	<a href="#">11</a>

Internet-Draft

TCP persist anomaly

July 2009

## 1. Introduction

According to [RFC 1122 \[RFC1122\] Section 4.2.2.17](#): 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. 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.

## 2. Discussion on [RFC 1122](#) Requirement

It needs to be emphasised that TCP MUST NOT take any action of its own when a particular connection is in persist state for a long time. As per [RFC 1122](#) as long as the ACK's are being received for window probes, it can continue to stay in persist mode. This is important because typically applications would want the TCP connection to stay open unless it explicitly closes the connection. For example take the case of user running a print job and the printer ran out of paper waiting for the user intervention. It would be premature for TCP to take action on its own. Hence TCP cannot act as a resource manager and it is the system or application's responsibility to take appropriate action.

At the same time, many existing TCP implementations that adhere strictly to the above verbiage of [RFC 1122](#), may fall victim to DOS attacks, if appropriate measures are not followed. For example, if we take the case of a busy server where multiple clients can advertise a zero forever (by reliably acknowledging the ZWP's), it could eventually lead to the resource exhaustion in the system. In such cases the system would need to take appropriate action on the TCP connection to reclaim the resources. The document is not intended to provide any advice on any particular resource management scheme that can be implemented to circumvent DOS issues arising due to the connections stuck in the persist state. The problem is applicable to TCP and TCP derived transport protocols like SCTP.

### 3. Description of Attack

If TCP implementations strictly follow [RFC 1122](#) and there is no instruction on what to do in persist condition, connections will encounter an indefinite wait. 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 its 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.

Applications on the sender can transfer all the data to the TCP socket and subsequently close the socket leaving the connection in orphaned state. If the application on the receiver refuses to read the data, the orphaned connection will be left holding the data indefinitely in its send queue.

If the above scenario persists for an extended period of time, it

will lead to TCP buffers and connection blocks starvation causing legitimate existing connections and new connection attempts to fail.

CERT is putting a advisory in this regard[VU723308] and is making vendors aware of this DoS scenario.

#### 4. Clarification Regarding [RFC 1122](#) Requirements

A consequence of adhering to the above requirement mandated by [RFC 1122](#) is that multiple TCP receivers advertising a zero window to a server could exhaust the connection and buffer resources of the sender. In such cases, and specially when the receiver is reliably acknowledging zero window probe, to achieve robustness, the system should be able to take appropriate action on those TCP connections and reclaim resources. A possible action could be to terminate the connection and such an action is in the spirit of [RFC 1122](#).

In order to accomplish this action, TCP MAY provide a feedback regarding the persist condition to the application if requested to do so or the application or the resource manager can query the health of the TCP connection which would allow it to take the desired action.

All such actions are in complete compliance of [RFC 793](#) and [RFC 1122](#).

## [5.](#) Conclusion

The document addresses the fact that terminating TCP connections stuck in the persist condition does not violate [RFC 1122](#) or [RFC 793](#). It also suggests that TCP MUST not abort any connection until either explicitly requested by the application to do so. The implementation guidelines of the request and the action are documented in [Section 7](#), and the details of mitigating the DoS attack are left to the

implementer.



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.

## [7.](#) Programming Considerations

To enable a server to clear connections in persist condition and reclaim resources, a socket interface needs to be defined. Note, this condition is mutually exclusive from a persist condition where we are not getting zero windows acknowledgement for the probes.

### PERSIST\_TIMEOUT

Format: `setsockopt(fd, SOL_TCP, PERSIST_TIMEOUT, persist_timeout_value)`

The interface allows applications to inform TCP that when the local connection stays in persist condition it can be cleared after a set time. Note that the default value of this option is indefinite.

TCP sender will save the current time in the connection block when it receives a zero window ACK. This time is referred to as the persist entry time. Thereafter every time the probe timer expires and before it sends another probe or an ACK carrying zero window is received a check will be done to see how long the connection has been in persist condition by comparing the current time to the persist entry time. If the timeout has been exceeded, the connection will be aborted.

Any time a ACK is received that advertises a non-zero window, the persist entry time is cleared to take the connection out of persist condition.

## 8. 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>".
- [VU723308] Manion, "Vulnerability in Web Servers <http://www.kb.cert.org/vuls/id/723308>", July 2009.

Internet-Draft

TCP persist anomaly

July 2009

#### 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

