TCP Maintenance and Minor Extensions
Working Group
Internet-Draft
Intended status: Informational
Expires: September 30, 2011

M. Bashyam
Ocarina Networks, Inc
M. Jethanandani
A. Ramaiah
Cisco
March 29, 2011

Clarification of sender behavior in persist condition.
draft-ietf-tcpm-persist-04.txt

Abstract

This document clarifies the Zero Window Probes (ZWP) described in Requirements for Internet Hosts [RFC1122]. In particular, it clarifies the actions that can be taken on connections which are experiencing the ZWP condition.

Table of Contents

1. Introduction

Section 4.2.2.17 of Requirements for Internet Hosts [RFC1122] says:

> "A TCP MAY keep its 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."
>
> DISCUSSION:
>
> > It is extremely important to remember that ACK (acknowledgment) 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 Zero-Window Probe (ZWP) mode is typically known as the 'persist condition'.

This guidance is not intended to preclude resource management by the operating system or application, which may request connections to be aborted regardless of them being in the persist condition, and the TCP implementation should, of course, comply by aborting such connections. TCP implementations strictly adhering to Section 4.2.2.17 of Requirements for Internet Hosts [RFC1122] have the potential to make systems vulnerable to Denial of Service (DoS) scenarios where attackers tie up resources by keeping connections in the persist condition, if such resource management is not performed external to the protocol implementation.

Section 3 of this document describes why implementations must not close connections merely because they are in the persist condition, yet must still allow such connections to be closed on command. Section 4 outlines a simple attack on systems that do not sufficiently manage connections in this state. Section 5 concludes with a requirements-language clarification to the RFC 1122 requirement.

2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

When used in lowercase, these words convey their typical use in common language, and they are not to be interpreted as described in Key words for use in RFCs [RFC2119].

3. Discussion on RFC 1122 Requirement

Per Requirements for Internet Hosts [RFC1122] as long as the ACK's are being received for window probes, a connection can continue to stay in the persist condition. This is an important feature because typically applications would want the TCP connection to stay open unless an application explicitly closes the connection.

For example take the case of user running a network print job during which the printer runs out of paper and is waiting for the user intervention to reload the paper tray. The printer may not be reading data from the printing application during this time. Although this may result in a prolonged ZWP state, it would be premature for TCP to take action on its own and close the printer connecting merely due to its lack of progress. Once the printer's paper tray is reloaded (which may be minutes, hours, or days later), the print job should be able to continue uninterrupted over the same TCP connection.

Systems that adhere too strictly to the above verbiage of Requirements for Internet Hosts [RFC1122] may fall victim to DoS attacks, by not supporting sufficient mechanisms to allow release of system resources tied up by connections in the persist condition during times of resource exhaustion. For example, if we take the case of a busy server where multiple (attacker) clients can advertise a zero window forever (by reliably acknowledging the ZWPs). This could eventually lead to the resource exhaustion in the server system. In such cases the application or operating system would need to take appropriate action on the TCP connection to reclaim their resources and continue to persist legitimate connections.

The problem is applicable to TCP and TCP derived flow-controlled transport protocols like SCTP.

Clearly, a system should be robust to such attacks and allow connections in the persist condition to be aborted in the same way as any other connection. Section 5 of this document provides the requisite clarification, in standards language, to permit such resource management

4. Description of one Simple Attack

To illustrate a potential DoS scenario, consider the case where many client applications open TCP connection with a HTTP [RFC2616] server, and each sends a GET request for a large page and stops reading the response partway through. This causes the client's TCP implementation to advertise a zero window to the server. For every large HTTP response, the server is left holding on to the response data in its sending queue. The amount of response data held will depend on the size of the send buffer and the advertised window. If the clients never read the data in their receive queues in order to clear the persist condition, the server will continue to hold that data indefinitely. Since there may be a limit to the operating system kernel memory available for TCP buffers, this may result in DoS to legitimate connections by locking up the necessary resources. 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.

A clever application might detect such attacks with connections that are not making progress, and could close these connections. However, some applications might have transferred all the data to the TCP socket and subsequently closed the socket leaving the connection with no controlling process, hereby referred to as orphaned connections. Such orphaned connections might be left holding the data indefinitely in their sending queue.

CERT has released an advisory in this regard[VU723308] and is making vendors aware of this DoS scenario.

5. Clarification Regarding RFC 1122 Requirements

As stated in Requirements for Internet Hosts [RFC1122], a TCP implementation MUST NOT close a connection merely because it seems to be stuck in the ZWP or persist condition. Unstated in RFC 1122, but implicit for system robustness, a TCP implementation MUST allow connections in the ZWP or persist condition to be closed or aborted by their applications or other resource management routines in the operating system.

An interface that allows an application to inform TCP on what to do when the connection stays in persist condition, or for application or other resource manager to query the health of the TCP connection is considered outside the scope of this document. All such techniques however are in complete compliance of TCP [RFC0793] and Requirements for Internet Hosts [RFC1122].

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

This document discusses one system security consideration as described in Security Considerations Guidelines [RFC3552]. In particular it describes a inappropriate use of a system that is acting as a server for many users. That and a possible DoS attack is discussed in Section 3.

8. 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, Ted Faber and David Borman for clarifying the objective behind this draft. To Wesley Eddy for his extensive editorial comments and to Dan Wing, Mark Allman and Fernando Gont on providing feedback on the document.

9. References

9.1. Normative 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.

9.2. Informative References

[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.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[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.

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
170 Tasman Drive
San Jose, CA 95134
USA

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

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

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