Internet Engineering Task Force (IETF)
Request for Comments: 6429
Category: Informational
ISSN: 2070-1721

M. Bashyam
Ocarina Networks, Inc.
M. Jethanandani
A. Ramaiah
Cisco

December 2011

TCP Sender Clarification for Persist Condition

Abstract

This document clarifies the Zero Window Probes (ZWPs) described in RFC 1122 ("Requirements for Internet Hosts -- Communication Layers"). In particular, it clarifies the actions that can be taken on connections that are experiencing the ZWP condition. Rather than making a change to the standard, this document clarifies what has been until now a misinterpretation of the standard as specified in RFC 1122.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6429.

Table of Contents

1. Introduction

Section 4.2.2.17 of "Requirements for Internet Hosts -- Communication Layers" [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 needs to be supported to prevent a connection from hanging forever if ACK segments that re-open the window are 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 that connections be aborted regardless of whether or not they are in the persist condition. The TCP implementation needs to, of course, comply by aborting such connections. If such resource management is not performed external to the protocol implementation, TCP implementations that misinterpret Section 4.2.2.17 of [RFC1122] have the potential to make systems vulnerable to denial-of-service (DoS) [RFC4732] scenarios where attackers tie up resources by keeping connections in the persist condition.

Rather than making a change to the standard, this document clarifies what has been until now a misinterpretation of the standard as specified in RFC 1122 [RFC1122].

Section 2 of this document describes why implementations might not close connections merely because they are in the persist condition, yet need to still allow such connections to be closed on command. Section 3 outlines a simple attack on systems that do not sufficiently manage connections in this state. Section 4 concludes with a requirements-language clarification to the RFC 1122 requirement.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Discussion of RFC 1122 Requirement

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

For example, take the case of a user running a network print job during which the printer runs out of paper and is waiting for the user to reload the paper tray (user intervention). 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 connection 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 needs to be able to continue uninterrupted over the same TCP connection.

However, systems that misinterpret Section 4.2.2.17 of [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, 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 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 maintain legitimate connections.

The problem is applicable to TCP and TCP-derived flow-controlled transport protocols such as the Stream Control Transmission Protocol (SCTP).

Clearly, a system needs to be robust to such attacks and allow connections in the persist condition to be aborted in the same way as any other connection. Section 4 of this document provides the requisite clarification to permit such resource management.

3. Description of One Simple Attack

To illustrate a potential DoS scenario, consider the case where many client applications open TCP connections with an 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 and therefore do not 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 starvation of TCP buffers and connection blocks, causing legitimate existing connections and new connection attempts to fail.

A clever application needs to 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 connections with no controlling process; such connections are referred to as orphaned connections. These orphaned connections might be left holding the data indefinitely in their sending queue.

The US Computer Emergency Readiness Team (CERT) has released an advisory in this regard [VU723308] and is making vendors aware of this DoS scenario.

4. Clarification Regarding RFC 1122 Requirements

As stated in [RFC1122], a TCP implementation MUST NOT close a connection merely because it seems to be stuck in the ZWP or persist condition. Though unstated in RFC 1122, but implicit for system robustness, a TCP implementation needs to 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 the persist condition, or that allows an 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 with TCP [RFC0793] and [RFC1122].

5. Security Considerations

This document discusses one system security consideration that is listed in "Guidelines for Writing RFC Text on Security Considerations" [RFC3552]. In particular, it describes an inappropriate use of a system that is acting as a server for many users. That use and a possible DoS attack are discussed in Section 3.

This document limits itself to clarifying RFC 1122. It does not discuss what can happen with orphaned connections and other possible mitigation techniques, as these are considered outside the scope of this document.

6. Acknowledgments

This document was inspired by the recent discussions that took place regarding the TCP persist condition issue in the TCP Maintenance and Minor Extensions (TCPM) Working Group mailing list [TCPM]. The outcome of those discussions was to come up with a document that would clarify the intentions of the ZWP as discussed in RFC 1122. We

would like to thank Mark Allman, Ted Faber, and David Borman for clarifying the objective behind this document. Thanks also go to Wesley Eddy for his extensive editorial comments and to Dan Wing, Mark Allman, and Fernando Gont for providing feedback on this document.

7. References

7.1. Normative References

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

[RFC1122] Braden, R., Ed., "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.

7.2. 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.

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[TCPM] IETF, "TCP Maintenance and Minor Extensions (tcpm) - Charter", <http://datatracker.ietf.org/wg/tcpm/charter/>.

[VU723308] Manion, A. and D. Warren, "TCP may keep its offered receive window closed indefinitely (RFC 1122)", November 2009, <http://www.kb.cert.org/vuls/id/723308>.

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: mjethanandani@gmail.com

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

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