

Network Working Group
Internet-Draft
Expires: August 3, 2004

A. Azcorra
C. Bernardos
I. Soto
UC3M
February 3, 2004

DoS vulnerability of TCP by acknowledging not received segments
draft-azcorra-tsvwg-tcp-blind-ack-dos-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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 August 3, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

TCP relies in communication peers to implement congestion control by hosts voluntary limiting their own data rate. Nevertheless this assumption introduces unsolved DoS attack opportunities.

A DoS attack can be easily performed by a host that acknowledges TCP segments not yet received (maybe even not sent).

This document presents and briefly describes the problem, already identified and pointed before, but also shows than it can be easily performed (with very interesting results) and proposes some server-side modifications to TCP stack in order to make this attack

Internet-Draft

TCP ACK DoS attack

February 2004

more difficult to perform.

Table of Contents

1.	Introduction	3
2.	Expeditious Blind ACK	4
3.	Implementation experimental results	6
4.	Proposed solution	7
5.	Security Considerations	9
	References	10
	Authors' Addresses	10
	Intellectual Property and Copyright Statements	12

Internet-Draft

TCP ACK DoS attack

February 2004

1. Introduction

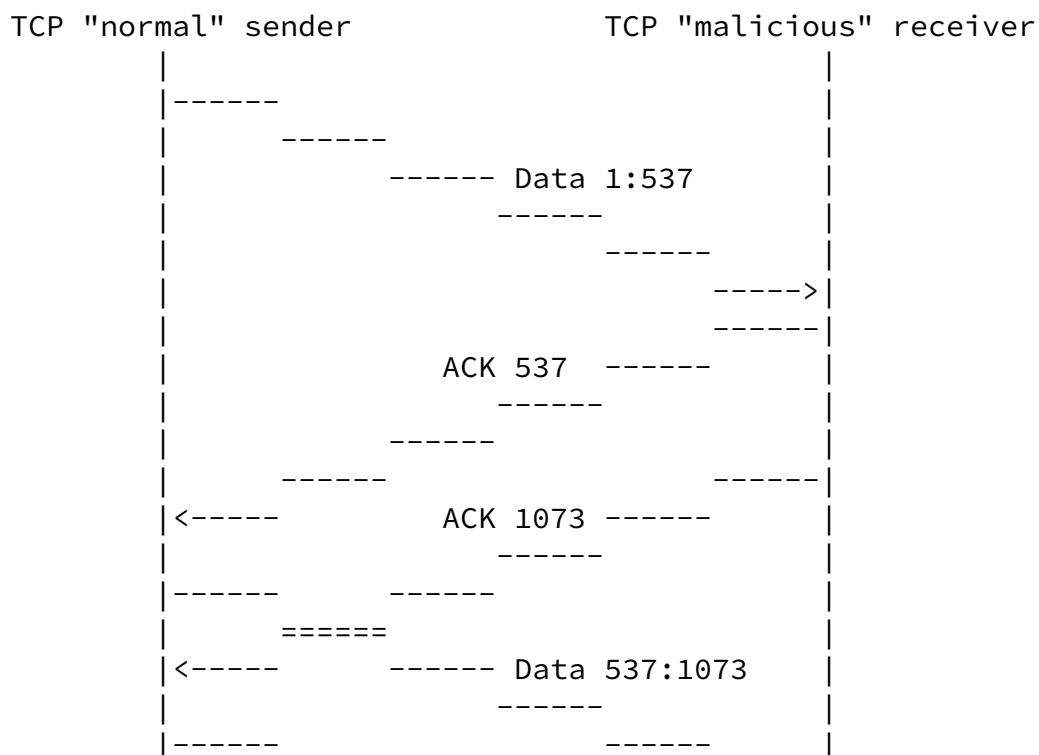
TCP Congestion Control, described in [RFC 2581](#) [[RFC2581](#)], relies in the peers involved in a communication. Therefore, hosts should voluntarily limit their own sending data rate. The problem is that TCP does not provide a mechanism to ensure that this bandwidth sharing on the Internet works. As is stated in [RFC 2581](#) [[RFC2581](#)], the Internet to a considerable degree relies on the correct implementation of the congestion control algorithms in order to preserve network stability and avoid network collapse. It is also pointed that an attacker could cause TCP endpoints to response more aggressively in the face of congestion by forging excessive duplicates acknowledgements or excessive acknowledgements for new data, driving a portion of the network into congestion collapse.

[SAVAGE] describes some of the these possible vulnerabilities of the TCP Congestion Control, not originated from bad implementations but for the TCP Congestion Control specification itself. These vulnerabilities can be exploited in order to obtain faster data incoming rate, but also to perform massive DoS attacks. One of these attacks, which is well known and it is pointed also in other documents (i.e. [[draft-nordmark-multi6-threats](#)]) is a especially dangerous attack. In this document we review this attack and prove it could be performed (by presenting results from some tests of a first implementation of an attacker), and we also provide a not very complex solution proposal, involving only changes on the server-side (attacked side) TCP stack.

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](#) [[RFC2119](#)].

2. Expeditious Blind ACK

Due to the nature of the TCP congestion control, a misbehaving receiver could acknowledge TCP segments not yet received, emulating this way a shorter round-trip time between sender and receiver. Because of the TCP's congestion window nature (linear growth with round-trip time), the sender will send data faster.



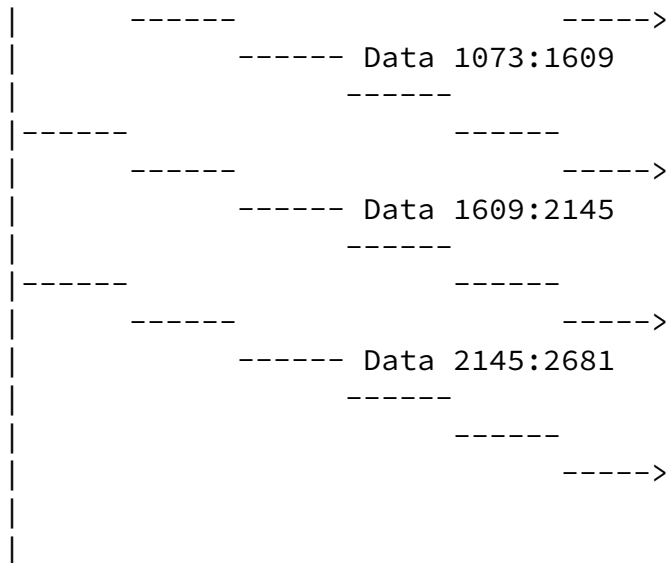


Figure 1: Expeditious Blind ACK sample flow

The attack is graphically shown in Figure 1. The "malicious" receiver sends ACK segments that acknowledges data yet not received. There are

some points that make this attack feasible:

- o It is easy to anticipate the sequence numbers to use in each ACK because senders generally send full-sized segments.
- o Most TCP implementations ignore received ACKs for segments that have not been sent yet. Therefore, it is easy to deploy some arbitrarily aggressive attacks (e.g. linear and exponential sequence number growths), even simultaneously.

[3.](#) Implementation experimental results

In order to prove that the attack described in this document is possible and very harmful, a small attacker tool (TerminETor) has been developed under a GNU/Linux system. This tool basically establishes a configurable number of simultaneous TCP connections (HTTP) to a server, and after the connection establishment, it starts a blind ACK generation, with configurable both linear and exponential growth. More parameters are also configurable, like the Receiver Maximum Segment Size (RMSS) specified by the sender during the connection establishment, the initial number of bytes acknowledged (usually the RMSS), the number of ACKs sent, and some more.

With a very preliminar version of this tool, we are able to, with a

56kbps client connection, make the sender generate about 25Mbps of outbound traffic. If this tool is used in a distributed way, a far more harmful DoS is possible.

More attacks and to different server architectures (O.S. and server implementations) should be performed in order to check if there is any O.S. that fixes this problem. Based on the results presented in [[SAVAGE](#)] and on the fact that the problem is not related to implementation bugs but to the TCP specification, we think that the problem is present in all TCP/IP stacks and that a solution should be provided and standardized.

Although this attack is well known today, it seems that there is a lack of proposed solutions to solve this problem without changing the client's TCP/IP stack. We have shown, by developing an attacker tool that the attack is possible and very harmful. Therefore, a solution involving only changes on the server side should be proposed, discussed and deployed.

[4.](#) Proposed solution

TCP Congestion Control [[RFC2581](#)] does not provide any mechanism to avoid misbehaving malicious TCP receivers perform DoS attacks. A sender has no means to be aware that the receiver has really received the packets it has acknowledged, so the sender increase its sending data rate based on the small RTTs calculated due to the fake ACKs

received.

TCP should provide a mechanism in order to assure that receivers have in fact received the data they acknowledge. This kind of solutions [[SAVAGE](#)] require changes in TCP/IP stacks of both clients and servers, which is a non realistic approach nowadays.

We propose a few changes, only on the TCP stack of the server side, that can greatly reduce the risk of suffering this kind of DoS attack. The proposed changes are the following:

1. A server SHOULD reset the TCP connection (i.e. send a RST) if it receives an ACK for data that the server has not sent yet.
2. A server SHOULD ignore ACKs that do not match the boundary of one of the segments it has sent.
3. A server SHOULD randomize segment boundaries (e.g. between 0.9 RMSS and RMSS).

The first modification is intended to disconnect an attacker that has sent acknowledgements in a too aggressive way (too fast), and its acknowledgements arrive to the sender before the corresponding data has been sent. This modification would force the attacker program to be more cautious in its acknowledgement rate, or to perform adaptative measurements to force the rate to the maximum, but without exceeding it. However, the attack would still be feasible.

The second modification intends to allow the sender distinguish between incoming "real" acknowledgements from "fake" ones. Real acknowledgements usually match segment boundaries, while fake ones may not when introducing modification 3. The reason to ignore the acknowledgements that do not match a boundary instead of resetting the connection, as done in modification 1, is that it is legal for a receiver to acknowledge part of a received segment. We have performed some experiments, and this appears to be a rare behaviour. Therefore, we believe it is an acceptable compromise to ignore the ACKs that do not match a boundary: a) For an attacker, this would force it to send much more ACKs for one to be accepted, making the attack more costly. b) for a correct TCP implementation, this would just cause an unnecessary retransmission. The situation would recover after the

retransmission because the elapsed time would allow the receiver data to drain, and the complete retransmitted segment would then be acknowledged.

The third modification is needed to make the second one effective. The objective is to make it difficult for an attacker to anticipate the sequence numbers to use in the ACKs. Currently, it is trivial for the attacker to anticipate the segment boundaries, as the sender will fill the segments up to SMMS as long as it has data to transmit, which is the case in a file download. With this modification, correct implementations would know the sequence number to acknowledge, while an attacker implementation would not.

These changes make this attack considerably more difficult without requiring changes in the client side (most of the Internet hosts). The implementation of the proposed changes on the server side is considered minor, and implies a low increase on the computational load of the protocol. The main modification is that the server has to maintain a list of the border sequence numbers sent, while currently is enough with a variable storing the sequence number of the highest data octet sent. Randomizing the size of the sent segment is not considered costly because the random distribution is not subject to cryptographical requirements, i.e. a simple pseudo-random technique would make the attack difficult enough.

Internet-Draft

TCP ACK DoS attack

February 2004

[5.](#) Security Considerations

This document presents some changes in order to improve security on the Internet, difficulting some DoS attacks that are possible now, without introducing any new attack opportunities.

Internet-Draft

TCP ACK DoS attack

February 2004

References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2581] Allman, M., Paxson, V. and W. Stevens, "TCP Congestion Control", [RFC 2581](#), April 1999.
- [RFC793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [SAVAGE] Savage, S., Cardwell, N., Wetherall, D. and T. Anderson, "TCP Congestion Control with a Misbehaving Receiver", ACM Communications Review 29(5), October 1999.
- [[draft-nordmark-multi6-threats](#)]
Nordmark, E. and T. Li, "Threats relating to IPv6 multihoming solutions",
[draft-nordmark-multi6-threats-00.txt](#) (work in progress),
October 2003.

Authors' Addresses

Arturo Azcorra
Universidad Carlos III de Madrid
Avda. Universidad 30
Leganes, Madrid 28911
Spain

Phone: +34 91 6248778
EMail: azcorra@it.uc3m.es
URI: <http://www.it.uc3m.es/azcorra/>

Carlos J. Bernardos

Universidad Carlos III de Madrid
Avda. Universidad 30
Leganes, Madrid 28911
Spain

Phone: +34 91 6248756
EMail: cjbc@it.uc3m.es
URI: <http://www.it.uc3m.es/cjbc/>

Azcorra, et al.

Expires August 3, 2004

[Page 10]

Internet-Draft

TCP ACK DoS attack

February 2004

Ignacio Soto
Universidad Carlos III de Madrid
Avda. Universidad 30
Leganes, Madrid 28911
Spain

Phone: +34 91 6245974
EMail: isoto@it.uc3m.es
URI: <http://www.it.uc3m.es/isoto/>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in [BCP-11](#). Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION

Azcorra, et al.

Expires August 3, 2004

[Page 12]

Internet-Draft

TCP ACK DoS attack

February 2004

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

