

Disabling PAWS When Other Protections Are Available
draft-nishida-tcpm-disabling-paws-00

Abstract

PAWS provides protection against old duplicated segments caused by wrapped sequence or earlier incarnated connections. One drawback of PAWS is that it requires to place timestamp option in all segments, which consumes 10-12 bytes in the option space of TCP. In addition, since PAWS just checks if timestamps is older or not, the protection logic is not very strong against malicious attacks or cannot work properly in some situations. On the other hand, some other technologies which can provide stronger protections than PAWS are becoming available these days. In this document, we propose to utilize other protection mechanisms as replacements of PAWS when they are available.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on December 22, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Conventions and Terminology	3
3.	Possible Mechanisms AS Replacements of PAWS	3
3.1.	TCP Increased Security (tcpinc)	3
3.2.	Multipath TCP	4
3.3.	TLS	4
4.	Duplicates from Earlier Connection Incarnations	5
5.	Security Considerations	5
6.	IANA Considerations	5
7.	References	5
7.1.	Normative References	5
7.2.	Informative References	6
	Author's Address	7

[1.](#) Introduction

PAWS (Protect Against Wrapped Sequences) defined in [[RFC7323](#)] is a technique that can identify old duplicate segments in a TCP connection or segments from earlier incarnated connections. PAWS utilizes timestamp option in TCP segments. When both TCP endpoints agree to use PAWS, all segments belong to this connection will have the options, which consumes 10-12 bytes of 40 bytes option space. As recent TCP connections use option space for other TCP extensions such as [[RFC2018](#)], [[RFC5925](#)] and [[RFC6824](#)], this feature tends to be considered as expensive these days.

Timestamp option is also used for RTTM (Round Trip Time Measurement). Gathering many RTT samples from the timestamp in every TCP segment may look useful approach to improve RTO estimations. However, some research results shows taking a few timestamps per RTT can be sufficient [[MALLMAN99](#)]. Also, some TCP implementations record the transmission time of each packet. In this case, timestamp option is not necessary to measure RTTs.

The basic idea of PAWS is that a received segment is considered as an old duplicate if the timestamp in it is less than the timestamps recently received on a connection. The timestamp values used in PAWS is 32-bit unsigned integers. Hence, when PAWS compares two timestamp values: t_1 , t_2 , it regards t_2 as "newer than t_1 " if $0 < (t_2 - t_1) <$

2^{31} , otherwise t_2 is considered as "older than t_1 ". This logic presumes timestamp is monotonically increased across connections, however, it can be confused in some cases such as where multiple nodes are behind the same NAT. In addition, if malicious attackers try to cheat the PAWS logic with random timestamp values, there will be 50% of chance for success on each try. This basically means PAWS can hardly contribute to securing TCP connections. On the other hand, several technologies which can be utilized for the same purpose are becoming available.

Based on these observations, we propose to utilize other mechanisms as replacements of PAWS when it is possible. The goal of the proposal in the draft is to provide stronger protections for old duplicated segments while facilitating the use of TCP option space by suppressing timestamp options. Another benefit of the proposal is that it can contribute to facilitating recycling of TCP connections in TIME_WAIT stat, which will be useful for busy servers.

2. Conventions and Terminology

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 [[RFC2119](#)].

3. Possible Mechanisms AS Replacements of PAWS

In this section, we present several possible mechanisms that can be used as replacements of PAWS. When these mechanisms are available and are activated in a TCP connection, PAWS can be disabled as it will be redundant. In order to disable PAWS protection, a simple signaling mechanism will be needed as this will require agreement on both ends. We discuss how to utilize these mechanisms and how to disable PAWS safely below.

3.1. TCP Increased Security (tcpinc)

Currently, TCP extensions that can provide unauthenticated encryption and integrity protection to TCP streams have been actively discussed in IETF [[TCPINC](#)]. As these proposed extensions is based on encryption algorithms, old duplicated segments in the same connection or segments from early connections with the same 4 tuples will easily be identified. In addition, it will be harder for malicious attacks to cheat this protections. To utilize this feature as a replacement of PAWS, we propose to update the encryption negotiation option (TCP-ENO) [[I-D.ietf-tcpinc-tcpeno](#)] by using 1 bit of global suboption in initial suboption byte. This bit indicates that the end point supports disabling PAWS. After endpoints confirm that both ends support disabling PAWS and encryption negotiation has been

successful, they will disable PAWS and will use timestamp options only for RTTM. However, in case TCP-ENO has failed or either side does not support disabling PAWS, PAWS MUST NOT be disabled if the use of TS option is negotiated.

3.2. Multipath TCP

Multipath TCP [[RFC6824](#)] can also be used for a replacement of PAWS. Multipath TCP maintains 64 bits sequence number space in the session and use DSS (Data Sequence Signal) option for this purpose in addition to the sequence number field in the TCP header. This DSS option can be served to provide the same protections as PAWS. By checking Data sequence number is DSS option, it can identify old duplicated segments. Because the data sequence number in the option should be exactly matched with the number stored in the session, MPTCP can provide stronger protection than PAWS. One way to signal disabling PAWS information is to use MP_EXPERIMENTAL option defined in [[I-D.ietf-mptcp-rfc6824bis](#)] during initial connection setup. Or, using 'B' bit in MP_CAPABLE option and extend MP_CAPABLE option to convey the info. Another requirement for disabling PAWS in an MPTCP session is that DSS mapping should be put in all data segments in the session. In case an MPTCP session falls back to TCP during SYN negotiation or either side does not support disabling PAWS, PAWS MUST not be disabled if the use of TS option is negotiated.

3.3. TLS

When TLS [[RFC5246](#)] is used for a TCP connection, old duplicated segments can be identified as segments in the connection are protected by encryption algorithms. One difficulty for using TLS as a replacement of PAWS is that it will be hard for TCP layer to know whether TLS is used or not. One possible way is to presume the use of TLS from port numbers (e.g. 443). Or, providing APIs to signal TCP layer can be another way although this will require to update applications. In addition, TCP needs to check if the other end supports disabling PAWS. In order for this, we will need to define a new TCP option in order to be used during SYN exchange. Another possibility is to utilize timestamp values in SYN segments in order to encode the feature negotiation information. An example of using timestamp value in SYN segments for feature negotiation is described in [[I-D.scheffenegger-tcpm-timestamp-negotiation](#)]. When either side does not support disabling PAWS, PAWS MUST not be disabled if the use of TS option is negotiated.

4. Duplicates from Earlier Connection Incarnations

As described in [[RFC7323](#)], the main purpose of PAWS is to protect against old duplicates from the same connection. In addition, it is expected to provide additional security against old duplicates from earlier connections. Although this feature is not strongly encouraged in the RFC, some implementations support it. We believe the protection described above can be used for this purpose as well. By using encryption logics or extended sequence number space, they can distinguish between the packets from the current connection and the packets from earlier connections. The protections will even be stronger than the protection PAWS can provide. We believe the replacements of PAWS will also be able to facilitate recycling the connections in TIME_WAIT. For example, it was reported some implementations replaced the connections in TIME_WAIT by a new incoming connection with the same 4 tuples when its timestamp is newer than the one in TIME_WAIT. However, this logic will not work properly when multiple clients behind NAT or when a node doesn't maintain global timestamp offset across all TCP connections. On the other hand, when a TCP connection can utilize the protections described above, it can recycle the connection in TIME_WAIT with more robust algorithms.

5. Security Considerations

TBD

6. IANA Considerations

This document may use the Experimental Option Experiment Identifier defined in [[RFC6994](#)]. In this case, an application for this codepoint in the IANA TCP Experimental Option ExID registry will be submitted.

7. References

7.1. Normative References

[I-D.ietf-mptcp-rfc6824bis]

Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", [draft-ietf-mptcp-rfc6824bis-11](#) (work in progress), May 2018.

[I-D.ietf-tcpinc-tcpeno]

Bittau, A., Giffin, D., Handley, M., Mazieres, D., and E. Smith, "TCP-ENO: Encryption Negotiation Option", [draft-ietf-tcpinc-tcpeno-18](#) (work in progress), November 2017.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", [RFC 6824](#), DOI 10.17487/RFC6824, January 2013, <<https://www.rfc-editor.org/info/rfc6824>>.
- [RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", [RFC 7323](#), DOI 10.17487/RFC7323, September 2014, <<https://www.rfc-editor.org/info/rfc7323>>.

7.2. Informative References

- [I-D.scheffenegger-tcpm-timestamp-negotiation] Scheffenegger, R., Kuehlewind, M., and B. Trammell, "Additional negotiation in the TCP Timestamp Option field during the TCP handshake", [draft-scheffenegger-tcpm-timestamp-negotiation-05](#) (work in progress), October 2012.
- [MALLMAN99] Allman, M. and V. Paxson, "On Estimating End-to-End Network Path Properties", Proceedings of the ACM SIGCOMM , September 1999.
- [RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", [RFC 2018](#), DOI 10.17487/RFC2018, October 1996, <<https://www.rfc-editor.org/info/rfc2018>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", [RFC 5925](#), DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC6994] Touch, J., "Shared Use of Experimental TCP Options", [RFC 6994](#), DOI 10.17487/RFC6994, August 2013, <<https://www.rfc-editor.org/info/rfc6994>>.
- [TCPINC] The IETF, "The TCPINC Working Group", <https://datatracker.ietf.org/wg/tcpinc/documents/>, 2014.

Author's Address

Yoshifumi Nishida
GE Global Research
2623 Camino Ramon
San Ramon, CA 94583
USA

Email: nishida@wide.ad.jp