Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2019

G. Fairhurst
T. Jones
R. Zullo
University of Aberdeen
October 19, 2018

Checksum Compensation Options for UDP Options
draft-fairhurst-udp-options-cco-00.txt

Abstract

This document describes a robust method for calculating checksums for use with UDP Options. The new method proposes an alternative checksum calculation for coverage of the option space. This is based on the IP checksum calculation, but uses an updated pseudoheader. The new method only checks the option portion of a UDP packet, but creates a checksum that compensates for the range of IP and UDP chekcsum validation methods that have been deployed, in this way the new method enhances the proability of NAPT traversal for packets that carry UDP-Options.

Status of This Memo

This Internet-Draft will expire on April 22, 2019.

Copyright Notice

Table of Contents

1. Introduction

UDP Options [I-D.ietf-tsvwg-udp-options] adds support for transport options in UDP [RFC0768]. When UDP is carried in IP two length fields describe the UDP datagram, the IP transport carries a payload length and the UDP header carries the length of the UDP datagram. In most datagrams currently forwarded by network devices the IP payload length is equal to the UDP length, UDP Options [I-D.ietf-tsvwg-udp-options] creates a surplus area by increasing the IP payload length while not varying the UDP length. Transport Options are then added in this surplus area in the form of a TLV encoded list.

The current specification for UDP permits sending datagrams with surplus data, but are not commonly observed, and many network devices assume that IP payload length is equal to UDP length and have used this value when calculating UDP checksums. This leads to the case where some middlebox devices (e.g. Firewalls, NAPT) and some endpoint implementations check or modify the UDP checksum in a way that leads to discard of UDP datagrams that carry UDP options.

This document describes common pathologies of network devices that incorrectly calculate the UDP checksum and proposes a new UDP Option to compensate for incorrect UDP checksum calculation.

2. 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. Middlebox Pathologies

Middleboxes and network interfaces can compute the UDP Checksum incorrectly in the presence of UDP Options based on the assumption that IP Payload Length and UDP Length coincide (an assumption that was equivalent before UDP Options).

These middleboxes use the IP Payload Length (obtained as IP Total Length - IP Header Length) to fill UDP pseudo-header Length field and also compute the checksum over the all IP Payload bytes.

This can lead to UDP Options packets that carry a correctly calculated checksum to be discarded by end-hosts or by middleboxes along the path.

Figure 1 shows UDP Checksum computation based on UDP Length and based on IP Payload Length and the fields that are different for the two calculation methods.

```
     Sum based on UDP Len   Sum basd on IP len      Delta between two

     +--------+--------+    +--------+--------+    +--------+--------+
     |     Source      |    |     Source      |    |                 |
     |     Address     |    |     Address     |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+
     |   Destination   |    |   Destination   |    |                 |
     |     Address     |    |     Address     |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+
     |  zero  |  PTCL  |    |  zero  |  PTCL  |    |        |        |
     +--------+--------+    +--------+--------+    +--------+--------+
     |   UDP Length    |    |IP Payload Length|    | Options Length  |
     +--------+--------+    +--------+--------+    +--------+--------+

     +--------+--------+    +--------+--------+    +--------+--------+
     |   Source Port   |    |   Source Port   |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+
     |    Dest Port    |    |    Dest Port    |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+
     |  UDP  Checksum  |    |  UDP  Checksum  |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+
     |   UDP Length    |    |   UDP Length    |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+

     +--------+--------+    +--------+--------+    +--------+--------+
     |                 |    |                 |    |                 |
     |                 |    |                 |    |                 |
     |   UDP Payload   |    |   UDP Payload   |    |                 |
     |                 |    |                 |    |                 |
     |                 |    |                 |    |                 |
     +--------+--------+    +--------+--------+    +--------+--------+

                            +--------+--------+    +--------+--------+
                            |                 |    |                 |
                            |   UDP Options   |    |   UDP Options   |
                            |                 |    |                 |
                            +--------+--------+    +--------+--------+
```

Figure 1: Checksum calculations

4. Checksum Compensation Option

This section introduces the Checksum Compensation Option (CCO), which suggests a new way to calculate the checksum for the option field.

The design of the CCO seeks to increase UDP Options compatibility with middleboxes and other existing network equipment, while at the same time providing error detection on UDP Options area in the same way that the UDP Checksum provides an integrity check for the UDP Header and UDP Payload.

CCO provides a checksum for UDP Option packets that is compatible with both variants of the checksum computation making the final value of the UDP Checksum computed on the whole IP Payload coincide with the the value that would be correctly computed soley on the UDP Length.

The Checksum Compensation Option (CCO) is the 2 byte one’s complement sum of the one’s complement sum of all 2 byte words in the UDP Options. Figure 2 describes the format of the CCO. The UDP Options area is divided into 2 byte words based on their alignment with the first byte of UDP packet (and not the first byte of UDP Options). This means that the first and the last byte of UDP Options can not preceded or be followed by another byte: in these cases the unpaired byte must padded respectively on the left and on the right with zero to form a 2 byte word.

```
+---------+--------+------------+
| Kind=xx | Len=4  |  Checksum  |
+---------+--------+------------+
  1 byte    1 byte    2 bytes
```

Figure 2: UDP CCO Option Format

[RFC0793] specifies: "The checksum field is the 16 bit one’s complement of the one’s complement sum of all 16 bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16 bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros." This method is equivalent to that specified for UDP [RFC0768].

The checksum also covers a 2 byte pseudo header conceptually prefixed to the UDP Options area. This pseudo header contains the length of UDP Options area. (The length also forms a part of the TCP and UDP pseudo field [RFC0793]).

Figure 3 shows the bytes on which CCO is computed and how, when present, the unpaired byte at the start and/or at end of Options area are included in the sum.

```
                               +--------+--------+
                               | Options  Length |
                               +--------+--------+
                               +--------+--------+
                               |                 |
                               |                 |
                               |   UDP Options   |
                               |                 |
                               |                 |
                               +--------+--------+

                               +--------+--------+
                               | Options  Length |
                               +--------+--------+
                               +--------+--------+
                               |  0x00  |        |
                               +--------+ - - - -|
                               |                 |
                               |   UDP Options   |
                               |                 |
                               |- - - - +--------+
                               |        |  0x00  |
                               +--------+--------+
```

Figure 3: UDP ECHORES Option Format

When this CCO checksum and the UDP Options field are covered by the UDP checksum calculation [RFC0768], the resulting UDP checksum value is numerically the same as when the UDP checksum calculation is calculated over only the UDP Payload. That is, the result retuned by both checksum computations Figure 1 coincide.

4.1. Calculating the CCO

The CCO can be present at any position within the Options space, the checksum field of the CCO MUST be aligned on a 2 byte boundary. This condition can be achieved by placing a NOP Option before the CCO in the case the number of bytes preceding the CCO (UDP Payload + UDP Options placed before CCO) is odd (see Figure 4).

```
                            +--------+--------+
                            |    UDP Header   |
                            |       ...       |
                            +--------+--------+
                            |   UDP Payload   |
                            |    and other    |
                            |   UDP Options   |
                            |       ...       |
                            |        +--------+
                            |        |  NOP   |
                            +--------+--------+
                            |  CCO   | CCOLen |
                            +--------+--------+
                            |    CCO  value   |
                            +--------+--------+
                            |Other UDP Options|
                            |       ...       |
                            +--------+--------+
```

Figure 4: Option Space layout

When calculated in this way, the CCO value is initialized to zero and the checksum is calculated over the UDP Options and the pseudo-header: the one’s complement of the result is then stored in the CCO field.

An alternative implementation could be to initialise the CCO field with the size of the UDP Options area (instead of initialising the CCO value to zero and combining with a pseudo header). This produces the same result, but allows the checksum to be performed using solely the UDP Options area.

4.2. Validating CCO

When a UDP packet containing CCO is received the Internet Checksum should be computed on the UDP Options area (2 byte aligned as described in Section 4.3) and the pseudo-header (the length of the received UDP Options), and the Options is valid if the one’s complement of the result is zero.

If the option checksum fails, all options MUST be ignored and any trailing surplus data (and Lite data, if used) silently discarded. UDP data that is validated by a correct UDP checksum MUST be delivered to the application layer, even if the UDP option checksum fails.

4.3. CCO Calculation Examples

This section provides examples of calculating the Checksum Compensation Option, similar to those presented in [RFC1071].

XXX IANA NOTE: The type of the CCO option has yet too be assigned, and may change. XXX

These examples use 204 (0xCC) as the type for the CCO option

In the first example the UDP Payload length is even and a MSS Option has been already placed in UDP Options area. CCO value is initialized with UDP Options Length (0x0008).

```
      UDP Length:                Even
      Preceding UDP Options:     MSS (kind 5, len 4, val 0x5c0)
      Following UDP Options:     None

      NOP Padding before CCO:    No
      Total UDP Options Length:  8

      UDP Options bytes 0/1:     0504
      UDP Options bytes 2/3:     05c0
      UDP Options bytes 4/5:     cc04
      UDP Options bytes 6/7:     0008
                                 ----
      Sum:                       d6d0

      CCO:                       292f
```

Figure 5: Checksum calculations

In the second example the UDP Payload length is odd and a MSS Option has been already placed in UDP Options area. The available space for CCO starts at an odd byte (NOP padding before CCO) and also UDP options space starts at odd byte (left zero padding of first byte). CCO value is initialized with UDP Options Length (0x0009).

```
      UDP Length:                 Odd
      Preceding UDP Options:      MSS (kind 5, len 4, val 0x5c0)
      Following UDP Options:      None

      NOP Padding before CCO:     Yes
      Total UDP Options Length: 9

      UDP Options bytes   0:      0005
      UDP Options bytes 1/2:      0405
      UDP Options bytes 3/4:      c001
      UDP Options bytes 5/6:      cc04
      UDP Options bytes 7/8:      0009
                                  ----
      Sum:                        9019

      CCO:                        6fe6
```

Figure 6: Checksum calculations

4.4. Interaction with other UDP Options

Interaction with other UDP Options

AE: Similarly to what happens with OCS, AE can be computed as if the AE hash and CCO value are zero. CCO value can be computed as if the CCO value is zero and after the AE hash has been computed.

ACS: The CCO has no interference with ACS since an ACS is computed only on UDP Payload bytes (no Header, no Options). The CCO value must be computed after the ACS has already been computed.

LITE: The CCO covers the entire UDP Option area, including any LITE option as formatted after swapping (or relocation) for transmission (or, equivalently, before the swap/relocation after reception). The CCO is computed after LITE swapping/relocation to guarantee the checksum compensation of the packet actually sent.

5. Acknowledgements

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI).

6. IANA Considerations

This memo includes no requests to IANA

7. Security Considerations

The security considerations for are described in [I-D.ietf-tsvwg-udp-options]. The proposed new method does not change the integrity protection offered by the UDP options method.

8. Normative References

[I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-05 (work in progress), July 2018.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC1071] Braden, R., Borman, D., and C. Partridge, "Computing the Internet checksum", RFC 1071, DOI 10.17487/RFC1071, September 1988, <https://www.rfc-editor.org/info/rfc1071>.

[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>.

Appendix A. Revision Notes

Note to RFC-Editor: please remove this entire section prior to publication.

Individual draft -00:

- Comments and corrections are welcome directly to the authors or via the IETF TSVWG working group mailing list.

- This update is proposed for WG comments.

Authors' Addresses

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen AB24 3UE
UK

Email: gorry@erg.abdn.ac.uk

Tom Jones
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen AB24 3UE
UK

Email: tom@erg.abdn.ac.uk

Raffaele Zullo
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen AB24 3UE
UK

Email: raffaele@erg.abdn.ac.uk