Introducing TCP Long Options by Invalid Checksum
draft-yourtchenko-tcp-loic-00

Abstract

This document discusses a possible approach to TCP option space expansion, which allows placing the long TCP options into the TCP SYN segments.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

What this document IS NOT: the definitive guide, the review of the
existing solutions, the full description of the option space upgrade,
the review of the existing solutions.
What this document IS: a write-up of an idea for a building block to be
used as part of a bigger protocol - intended for information purposes
and further development only - verifying the feasibility of this
approach will need extensive experiments.

2. Notational Conventions

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. Problem Statement

The TCP options are the mechanism which is used to evolve the TCP
protocol - thanks to their existence, such additions as Selective
Acknowledgements and Timestamps [refs tbd] (to name just a couple) were
introduced. However, the space in the TCP segment where the options may
be transmitted is very scarce and is currently at the borderline of
being exhausted. This practically blocks any further development in the
TCP protocol space.
In order for the TCP development to continue, we need a way to dedicate
a larger storage space for TCP options within the segment - further
referred as TCP Long Options.
A mechanism for doing that must fulfil at least two prerequisites:
transmission of the Long Options within the TCP SYN, and a graceful
fallback to the 'legacy' mechanism in case the Long Options can not be
used (thus abandoning all of the Long Options).
The above two are to certain extent at odds with each other - the SYN
is the very first segment within the TCP session, so any drastic change

that is misunderstood by the remote party or the middleboxes will cause
the segment to be dropped, thus interrupting the communication.

4. Proposed High-level Approach

The proposal is to transmit two SYN segments instead of one, created
with different goals in mind: the first one, aimed at backwards
compatibility, would merely signal the sender's desire to use Long
Options. The second SYN segment, aimed at the parties that understand
the new mechanism, would contain the Long Options themselves. The Long
Options would be in place where there normally is data (there's simply
no other space) - so this segment, together with the first one, would
form a contradiction from the perspective of the TCP protocol, if
interpreted by the unaware node.
To mitigate the above issue, we can explicitly mark the second SYN as
"TCP-invalid". The simplest way to do this is to increment the valid
TCP checksum by 2. With such a modification, the second packet will be
either dropped by a middlebox that does not support the new method, or
by the destination node itself - if the destionation does not support
this method.
On the other hand, if the stack supports the new method for the Long
Options, it can treat the first segment as a partial duplicate of the
second - thus allowing to upgrade the protocol behaviour. (NB: the
overall upgrade protocol is a much larger problem and is out of scope
for this document).

5. Implementation Details

Allocate two new TCP options: a 4 bytes long "LOIC-FLAG" and a 4 bytes
long "LOIC-LEN" (Long Options with Invalidated Checksum).
The first one is aimed into inclusion into the 'compatible segments' to
signal that the system understands the long options mechanism, as well
as later possibly be used as a signaling mechanism about the end-to-end
connectivity for the Long Options segments. The goal here is to
minimize the potential of the disruption for the existing applications.
The two bytes of usable payload of the second option will hold the
length of the TCP Long Options area (zero being an allowed value and
meaning there is no Long Options at all). The TCP Long Options area
will be placed between the end of the 'classic' TCP header and the
beginning of the TCP segment data. The presence of this option will
mean that before verifying the TCP checksum, one MUST decrement the
received value by 2 - and only then verify the checksum. To allow more
efficiency in the implementations, this option MUST be the first TCP
option within the segment - this way the analysis of the TCP checksum
would not be impacted too much.

6. Security Considerations

[[Placeholder.]]

7. Acknowledgements

The concept of using the invalid checksum came up during one of the discussions with Dan Wing about an unrelated matter. Thanks to following individuals for the initial discussions about this idea: Wesley Eddy, Alan Ford, Anantha Ramaiah, Fernando Gont [[Please remind me who I forgot, I forgot a few folks for sure]]

8. IANA Considerations

This document, being informational, has no IANA actions. However, its derivatives that adopt the described approach, would have an action to update the registry of the TCP options to include the LOIC-FLAG and LOIC-LEN definitions.

9. References

9.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informational References

[RFC1671]	Carpenter, B., "IPng White Paper on Transition and Other Considerations", RFC 1671, August 1994.

Author's Address

Andrew Yourtchenko Yourtchenko Cisco Systems, Inc. De Kleetlaan, 7 Diegem, B-1831 Belgium EMail: ayourtch@cisco.com