Transport WG
Internet-Draft
Intended status: Experimental
Expires: July 25, 2013

A. Knutsen
Blue Coat Systems
A. Ramaiah
NTT MCL Inc
A. Ramasamy
Cisco

January 21, 2013

TCP option for transparent Middlebox negotiation
draft-ananth-middisc-tcpopt-01.txt

Abstract

This document describes a TCP option for use by middleboxes to facilitate transparent detection of other middleboxes along the path of the TCP connection during the connection initiation phase. The option has no effect if an appropriate middlebox is not present on the path.

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 http://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 July 25, 2013.

Table of Contents

1. 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]

Middlebox

"Middleboxes: Taxonomy and Issues" [RFC3234] defines a middlebox as follows:

"A middlebox is defined as any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host."

Proxy

HTTP1.1 [RFC2616] defines a proxy as follows: "An intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients." Proxies exist for many protocols, such as HTTP, CIFS, MAPI and streaming. Since they act as both server and client, they have separate TCP connections to the original client and the actual server (also referred to as the "Original Content Server"). Proxies are often implemented on middleboxes. Proxies fall into two general categories: "Explicit" and "Transparent". The client must be configured to connect to an explicit proxy; it then passes the server address to it using an application protocol, such as HTTP. Transparent proxies require no client configuration; they intercept the client connection to the server, speaking to the client on its behalf, and make a separate connection to the server without the knowledge of the client. This memo deals exclusively with cooperating transparent proxies.

Peer

Two or more middleboxes with an effective association or relationship are peers. For example, one middlebox might compress data while another middlebox decompresses; neither middlebox can correctly manipulate traffic unless they both establish the existence of the other and coordinate their actions. The most common peering relationships are two-way (like the compression example) in which one middlebox performs a transformation that the other middlebox inverts. However, two-way peering that does not involve an invertible transformation is also found, as are n-way peering relationships where the "intermediate" peers are simply operating in a pass-through or failover mode.

2. Introduction

For middleboxes that operate on TCP-based application protocols, it is highly desirable for negotiation information to be carried within packets containing valid TCP protocol data. One significant kind of service offered by such middleboxes is application acceleration, in which there is an intrinsic requirement to be efficient and avoid network round trips. A middlebox-negotiation mechanism that imposes additional round trips could defeat the purpose of such middleboxes. Middlebox negotiation on a per-connection basis allows for significant advantages in scale and flexibility. Middlebox-based services can be invoked dynamically, using the appropriate peers, without any need for statically identifying middleboxes to end hosts or identifying middleboxes to each other. Dynamic negotiation means that there is no overlay routing required for middlebox-based services, where such overlay routing could potentially clash with the underlying IP routing. For all of these reasons, multiple vendors have implemented TCP-option-based negotiation mechanisms; and for similar reasons, this memo requests a TCP option for middle box negotiation.

The TCP middlebox negotiation option (TCP-MNO) allows a source node (initiating middlebox) on the initiating path of a TCP connection to request a response from other middleboxes with a matching capability closer to the destination host. In addition, it allows the initiating middlebox to provide information to the other middleboxes which they may need to decide whether to respond to this request. A middlebox MAY examine TCP packets with the SYN bit set to determine if the associated TCP connection qualifies for the middlebox-provided service. If so, the middlebox MAY insert the TCP-MNO into the packet header before sending it on. A middlebox MAY examine packets containing the TCP-MNO to determine if the associated TCP connection qualifies for the middlebox-provided service. If so, the middlebox MAY take additional actions to coordinate with the initiating middlebox. Such actions MAY include acknowledging the SYN packet to intercept the connection; originating a separate connection to the client; or perhaps notifying a management station.

TCP end hosts are unaware of TCP-MNO, and its usage is strictly for use by TCP middleboxes. Note however that the middlebox may be running on the TCP end host itself in some form, and may use TCP-MNO. Multiple vendors of WAN optimization products have used similar (but incompatible and proprietary) mechanisms since at least 2004. Those existing vendor systems use multiple TCP option code points, all of which are officially unassigned by IANA. The goal of this memo is to standardize a single TCP option code point for this functionality. Note that TCP is inherently end-to-end, and the network infrastructure (including middleboxes) are not supposed to alter the

TCP headers, but the current deployment practice has resulted in the middleboxes violating this architectural principle to address a market need. Goal of this memo is not to say this is good or bad but merely acknowledges the reality and encourages the use of a shared documented option number for such uses.

This memo is the product of discussion among some of the vendors currently using incompatible proprietary TCP options for middlebox negotiation. It is a non-goal of this memo to achieve interoperability of middlebox negotiation between multiple vendors. It needs to be noted that an earlier document [I-D.knutsen-tcpm-middlebox-discovery], is re-written as the current document after agreement from multiple vendors.

3. Requirements of the TCP-MNO

The following are the requirements of TCP-MNO :

1) The TCP-MNO MUST be variable length to accommodate multiple vendor option formats.

2) The TCP-MNO MUST have a vendor ID which can identify the specific vendor as implied by 1)

3) The TCP option space limitation puts a burden on how flexible the option can be. Please refer section 6 below.

4) TCP option numbers already in use by proprietary systems SHOULD NOT be reused for TCP-MNO since it would create confusion. (These option numbers would get eventually retired when all vendors migrate to the newly allocated TCP-MNO option)

5) The TCP-MNO SHOULD be used for middleboxes only. The hosts are expected to silently ignore this option.

4. Description

The TCP-MNO MAY be included in the TCP handshake (SYN and SYN+ ACKpackets). The TCP-MNO contained in the SYN packet is used to discover peer middleboxes along the path to the server. The TCP-MNO option MAY be present in the TCP SYN+ACK and other TCP data packets as well.

It should be noted that a common use of middleboxes is to set up cooperating peer proxies. One example is to implement a feature which optimizes the WAN traffic, like a compression protocol. In these cases, the option is used by the device nearer to the client to discover a possible device nearer to the server. Thus the client and server application are not aware of the option. Figure 1 illustrates the text above.

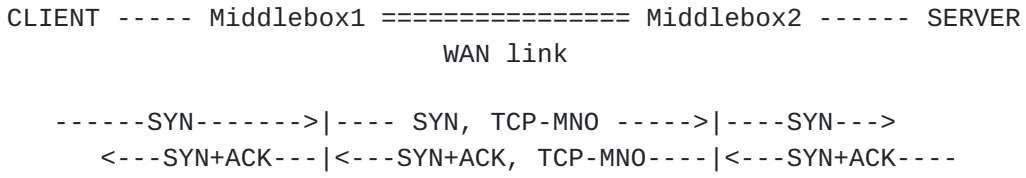
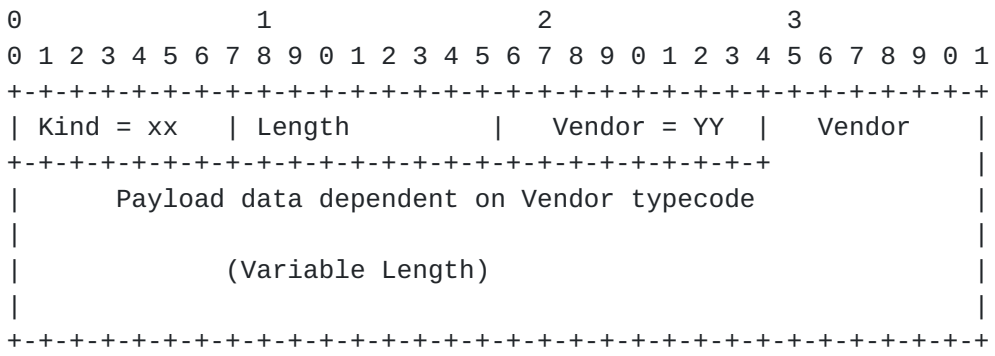


Figure 1: TCP-MNO option insertion during TCP handshake

5. TCP-MNO option format

The following is the agreed-upon option format for TCP-MNO.



(One tick mark represents one bit)

Figure 2 : Format of TCP-MNO

Even after 8 years of commercial development, there are only a handful of vendors with different option-based negotiation schemes. Accordingly, 8 bits is likely to be more than sufficient for vendor identification. Ideally, future work would define a single interoperable middlebox negotiation scheme to replace the multiple vendor-specific schemes.

The PEN (32 bit Private Enterprise numbers) is an existing mechanism which could be used to identify various vendors (http://www.iana.org/assignments/enterprise-numbers). Because of the very limited amount of TCP option space, using 32 bits for an enterprise number is impractical. The next section talks more about the TCP option space issue.

Other formats for TCP-MNO have been previously considered, including an approach in which the option contains a Type code followed by Vendor ID. That design had the advantage of supporting an extension into 32-bit enterprise numbers. But no meaningful commonality could be identified among the different vendor schemes; at any level beyond the use of a TCP option, each functions quite differently and there was no clear basis for choosing a standardized type scheme.

6. TCP option interoperability

This section touches upon the interoperability issues of TCP-MNO with other TCP options.

It needs to be noted that the TCP-MNO can be inserted only if the TCP option space permits. Given that there is a maximum of 40 bytes for TCP options, A SYN (with MSS, window scale, SACK permitted, and timestamp options) leaves 16 bytes spare (if the options are word-aligned) or 21 bytes spare (if the options are not word-aligned) which may just make it. In particular, it is not typically possible to insert TCP-MNO if a vendor-proprietary option has already been inserted, nor vice-versa. Strictly speaking, this conflict is not a new problem but the attempt at standardization may mean it occurs more often. It is already true in most cases that it is not possible to insert vendor A's option if vendor B's option has previously been inserted. This memo does not attempt to define rules on precedence or priority of such options, nor does it define circumstances in which a proprietary option may safely be replaced by TCP-MNO.

Multipath TCP ([I-D.ietf-mptcp-multiaddressed]) would have more options sent in the SYN, therby restricting the TCP option space that can be used for TCP-MNO.

The TCP-MNO option is incompatible with TCP options which aim to protect the integrity of the TCP SYN. An example of such an option is the TCP MD5 signature option[RFC2385]. Such TCP options are not commonly seen by current WAN optimization systems, so the restriction should not pose any worries.

7. IANA Considerations

This section is interpreted according to [RFC5226])

This document needs a new TCP option to be allocated by IANA for the TCP-MNO option from the "TCP Option Kind Numbers" registry maintained at http://www.iana.org.

This document also defines an 8 bit middle-box vendor id field, for which IANA is requested to create and maintain a new sub-registry entitled "Middle box vendor id" under the new middle box TCP option. The middle box vendor id will be used to differentiate multiple vendors. Initial values for the middle box vendor id sub-registry are given below; future assignments are to be made on a "First Come First Served" basis.

ID	Middle-box Vendor
0	Reserved
1	Bluecoat
2	Riverbed
3	Cisco
4-127	Unassigned
128-255	Reserved

It is envisioned that the reserved codes can be used in the following manner in future. Exact definitions of them is not spelled out here and it is left to such a time when there is more clarity around requirements and usage model.

1) Standard code: Standard code can be used in a such way that multiple vendors recognize the same and are interoperable for negotiation purposes

2) Extended vendor id: Should the unassigned space run out in the future, either the reserved space can be opened up for assignment, or a speical id can be added to extend the vendor id space to more than one byte (say by including OUI after the 1 byte vendor id)

It needs to be noted that at the time of writing this document, the following vendors have used the following TCP option numbers for TCP middlebox auto-discovery.

Vendor	TCP option number
Bluecoat	253
Cisco	33
Riverbed	76 and 78

Figure 3: Current TCP auto-discovery option numbers used by various vendors.

8. Security Considerations

The TCP-MNO option is incompatible with any TCP option which aims to protect the integrity of the TCP SYN, such as the TCP MD5 signature option[RFC2385] and TCP-AO option[RFC5925]. In addition, this memo's approach to middlebox negotiation introduces two security issues. First, the option-based paradigm intrinsically involves modifying traffic. Some traffic may be modified even though it cannot or should not receive the middlebox-based service. Secondly, there is no authentication mechanism defined in this memo to ensure that middleboxes communicate only with well-behaved and trusted potential peers. A rogue middlebox can potentially insert itself into a functioning community of middleboxes, disrupting the service provided.

9. Contributors

The following individuals contributed immensely to this document :

Ron Fredrick, Blue Coat.

10. Acknowledgements

Thanks to Lars Eggert for forming the middisc alias which was responsible for discussions leading to the current document. Thanks to Wes Eddy and David Harrington for the constant encouragement in getting this document written.

11. References

11.1. Normative References

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

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

11.2. Informative References

[I-D.ietf-mptcp-multiaddressed] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", draft-ietf-mptcp-multiaddressed-12 (work in progress), October 2012.

[I-D.knutsen-tcpm-middlebox-discovery] Knutsen, A., Frederick, R., Mahdavi, J., Li, Q., and W. Yeh, "TCP Option for Transparent Middlebox Discovery", draft-knutsen-tcpm-middlebox-discovery-04 (work in progress), May 2010.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[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.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

Authors' Addresses

Andrew Knutsen
Blue Coat Systems
420 North Mary Ave
Sunnyvale, CA 94085-4121
USA

Phone: +1 (408) 220-2250
Email: andrew.knutsen@bluecoat.com

Anantha Ramaiah
NTT MCL Inc
101 S. Ellsworth Avenue, Suite 350
San Mateo, CA 94401
USA

Email: ananth@nttmcl.com

Arivu Ramasamy
Cisco
170 Tasman Drive
San Jose, CA 95134
USA

Phone: +1 (408) 525-5962
Email: mani@cisco.com