TCP Option for Transparent Middlebox Discovery
<draft-knutsen-tcpm-middlebox-discovery-03.txt>

Status of this Memo

Distribution of this memo is unlimited.

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire February, 2010.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document describes a TCP option intended to facilitate transparent detection of middleboxes (or services playing that role) along the path of a TCP connection as the connection is made. The option has no effect if an appropriate middlebox is not on the path.

Table of Contents

1. Terminology .................................................... 2
2. Introduction .................................................... 3
3. Survey of Existing Technology ................................. 4
   3.1. LAN Discovery Protocols .................................. 4
   3.2. IP-based protocols ....................................... 4
   3.3. Resource Reservation / QoS Protocols .................... 4
   3.4. Requirements Documents .................................. 4
4. Conventions .................................................... 4
5. Operation ....................................................... 5
   5.1. Option Format ............................................ 5
   5.2. Initiating Discovery Request .............................. 6
   5.3. Responding to Discovery Request ......................... 6
   5.4. Reserved Option Values .................................. 7
6. Interoperability Issues ......................................... 7
7. Programming and Manageability Considerations ............... 7
   7.1. TCP User Interface ..................................... 7
8. Security Considerations ....................................... 8
9. IANA Considerations .......................................... 8
10. Acknowledgments .............................................. 8
11. References .................................................... 8
   11.1. Normative References .................................. 8
   11.2. Informative References ................................ 9

1. Terminology

Client

This is the original initiator of a request. The request is generally directed to a server.

Server

A host providing services to clients.

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.

Tunnel

A Tunnel can be viewed as two middleboxes (or software acting in that role) acting in concert to provide a service, such as security or compression. They will generally create a TCP connection between themselves, in addition to the client and server connections.

2. Introduction

The TCP Transparent Middlebox Discovery option is intended to allow a node on the initiating path of a TCP connection to request a response from middleboxes with a particular capability closer to the destination host. In addition, it allows the source node to provide information to the middlebox which it may need to decide whether to respond. The response may take the form of acknowledging the SYN packet and intercepting the connection or some other response, such as originating a separate connection to the client, or perhaps notifying a management station.

While there are numerous other technologies related to resource
discovery, there are several specific requirements which have led a number of products to pursue the approach outlined in this specification. Middleboxes which perform transparent interception are often inserted in the path by means of layer 4 redirection. For middleboxes which operate on TCP-based application protocols, this means that it is highly desirable for discovery information to be carried within packets containing valid TCP protocol data. In addition, one significant class of service offered by such middleboxes is application acceleration; solutions which impose additional round trips may defeat the purpose of such middleboxes. Section 3 considers a number of existing discovery protocols and their potential suitability for transparent middlebox discovery.

3. Survey of Existing Technology

3.1. LAN Discovery Protocols

These protocols, such as the Service Location Protocol [RFC2608], Link-Local Multicast Name Resolution [RFC4795], and Universal Plug and Play over UDP HTTP [HTTPU] [SSDP], are unsuited to the purpose of this option because they are limited to LAN scope (or require multicast infrastructure).

3.2. IP-based protocols

IP-based protocols, such as the ICMP ECHO request [RFC792], are not suitable for two reasons: they may not follow the TCP connection path if there is layer 4 redirection (such as WCCP [WCCP]) taking place; and they require an extra round trip time.

3.3. Resource Reservation / QoS Protocols

The NSIS framework [RFC4080] solves a similar problem. However it also adds delay, and may not work in the presence of L4 redirection.

3.4. Requirements Documents

"Requirements for Discovering Middleboxes" [LEAR01] discusses requirements for a class of problems similar to the one addressed here.

4. 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].
5. Operation

The Middlebox Discovery Option is implemented as a TCP Option. This TCP Option is included in the TCP connection handshake. A Request option to discover middleboxes is sent in the TCP SYN packet, and a Response option MAY be present in the TCP SYN-ACK packet.

It should be noted that a common use of middleboxes is to set up tunnels, for example to implement a compression protocol. In these cases, the option is used by the device nearer the client to discover a possible device nearer the server. Thus, the client and server applications are not aware of the option.

5.1. Option Format

![Option Format Diagram](https://example.com/option_format.png)

(One tick mark represents one bit.)

Figure 1: Format of the Middlebox Discovery Option

The "R" bit indicates whether this option is a Request or Response. If the bit is 0, the option is a Request; otherwise it is a Response. Some TCP implementations reflect unknown options received in a TCP SYN back in their SYN-ACK response. The "R" bit can be used to distinguish these invalid responses from actual Middlebox Discovery Responses.

The Device Capability identifies the targeted devices. The "P" bit indicates if the device capability is defined by IANA or by another organization. If the bit is 1, a 3-byte IEEE Organizational Unit Identifier (OUI) indicating which organization defined the capability MUST follow the Device Capability field.

If the option length is greater than the total length of the option kind, length, device capability and optional OUI, the remaining data ("target data") MUST be interpreted according to the device capability. The expected use for this data is to allow the targeted
The device capability specifies what target data MUST be present in
the option, if any.

Requests are only valid in SYN packets. They MUST NOT appear in other
segments and MUST be ignored when found outside of a SYN.

5.3. Responding to Discovery Request

Requests received in any state except SYN-SENT MUST be ignored.

Devices MUST NOT respond to requests which have not been validated
using the target data, if required by the device capability.

Responses MUST have the "R" bit set to 1.

The device capability specifies what target data MUST be present in
the option, if any.

Responses are only valid in valid SYN-ACK packets. They MUST NOT
appear in other segments and MUST be ignored when found outside of a
SYN-ACK.

All further transactions on the connection are outside the scope of
this document.
5.4. Reserved Option Values

Device capabilities with the P (private) bit set to 0 are reserved for assignment by the IANA.

If the "P" bit is set to 1, a 3-byte IEEE Organizational Unit Identifier follows the device capability. In this case, this ID defines the interpretation of the device capability, providing each organization with its own private device capability space.

6. Interoperability Issues

TCP options generally are not preserved when a proxy or tunneling device re-originites a connection. Some firewalls also strip TCP options. Discovery Requests and Responses cannot be expected to traverse such devices.

Implementers should be aware that in some cases packets originated by a middlebox may be routed back through it. If a middlebox can both accept incoming Middlebox Discovery Options and generate outgoing Middlebox Discovery options, it is important that some measures be taken to prevent interception of connections initiated by oneself. This can be accomplished either explicitly (via data included within the Middlebox Discovery Option that identifies the middlebox) or implicitly (via the middlebox maintaining a table of all connection 4-tuples it has originated so as to not re-intercept them).

There may be situations where other options are required in the SYN packet which do not leave enough room for all of the target data necessary for the desired device capability to be advertised. In these cases, a shorter alternate device capability may be defined which signals the request to further negotiate the capability after the handshake completes. This will impact performance by introducing an extra round-trip during connection set-up, but this may be the only way to perform the negotiation within the limited TCP option space available.

7. Programming and Manageability Considerations

Network analysis tools and firewalls MAY interpret this option for management purposes.

If this option is detected by an application which is not prepared to interpret it, it MUST be ignored.

7.1. TCP User Interface

RFC 793 [RFC793] defines an example user interface for TCP,
consisting of active and passive OPEN, SEND, RECEIVE, CLOSE, STATUS and ABORT commands. This option affects the following commands:

The active OPEN command MUST be augmented so the device capability and target data can be specified, if requests will be sent.

The passive OPEN command MAY be augmented so the response can be specified. However a different API, where the OPEN returns before the response is sent, and an additional OPEN-RESPONSE command is used to complete the open, may be more suitable. This API is outside the scope of this document.

The STATUS command MAY be augmented to return the response data.

8. Security Considerations

Since this option is in the TCP header, it will be protected by IP Security [RFC4301], the MD5 Signature option [RFC2385], and the TCP Authentication Option [TCP-AO]. IP Security with privacy will prevent detection of the option; all may prevent responses.

When transport-level security, such as TLS [RFC5246], is used, the option will be visible. The "target data" MAY be separately protected, as defined by the device capability.

9. IANA Considerations

This section is to be interpreted according to [RFC5226].

This document defines a new namespace of standard discoverable device capabilities (when the "P" bit is set to 0). This space is 14 bits wide. It is expected that this namespace will be administered by the IANA.

IANA will need to allocate a new 8-bit TCP option number for this option from the "TCP Option Kind Numbers" registry maintained at http://www.iana.org.

10. Acknowledgments

11. References

11.1. Normative References


11.2. Informative References


[LEAR01] Lear, E., "Requirements for Discovering Middleboxes", <draft-lear-middlebox-discovery-requirements-00.txt>, April 2001

Authors’ Addresses

Andrew Knutsen  
Tel: (408) 220-2250  
andrew.knutsen@bluecoat.com

Ron Frederick  
Tel: (408) 220-2006  
ron.frederick@bluecoat.com

Jamshid Mahdavi  
Tel: (408) 220-2313  
jamshid.mahdavi@bluecoat.com

Qing Li  
Tel: (408) 220-2369  
qing.li@bluecoat.com

Wei Jen Yeh  
Tel: (408) 220-2098  
weijen.yeh@bluecoat.com

Blue Coat Systems Inc.  
420 North Mary Ave.  
Sunnyvale, CA 94085-4121