Network Working Group
Internet Draft

Bellovin, Buchsbaum, Muthukrishnan
AT&T Labs--Research

Expiration Date: April 2000

October 1999

TCP Filters

draft-bellovin-tcpfilt-00.txt

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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."

This draft document will be submitted to the RFC Editor as an Experimental RFC. Distribution of this document is unlimited.

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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

2. Abstract

We propose a method to specify general-purpose filters between TCP and the application layer. The method is incrementally deployable, as neither party will use a filter layer without the other's consent.

3. Introduction

TCP [RFC793] provides applications with a simple reliable byte stream. While this is a powerful primitive, some applications require more. For example, many applications require delimited records. Others benefit from compression or encryption. While these services can be implemented by individual applications, it is sometimes desirable to provide a common mechanism. This is especially useful for administratively decreed options that are intended to be used by unchanged applications. For example, an administrator can configure a pair of systems so that all FTP traffic between the two is compressed, without modifying either the clients or the servers.

We describe a simple mechanism based on TCP options. During the exchange of SYN packets, one side initiates a filter negotiation by announcing what filters it is prepared to employ. The other responds in the next ACK packet by listing the subset of those filters that it is prepared to accept. Thence, each side applies the agreed upon filters to the payload in each subsequent packet.

For the sake of simplicity, there is no negotiation after the initial three-way handshake. Furthermore, both directions use the same set of filters.

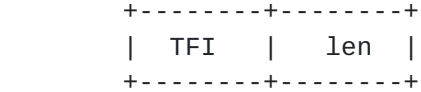
3.1. Specification of Requirements

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 RFC 2119 [RFC-2119].

4. Option Format

Each filter uses one TCP option, of type TFI (to be assigned by IANA). SYN packets contain a list of filter options; these represent the types of filter the sender wishes to receive.

A filter option has the following format:



A filter option announcement can have parameters; parameters are option-dependent and are described in the appropriate documents.

Each side includes the agreed upon options until it is guaranteed that the other side has received an ACK with those options. (Cf., Protocol.) Thence no filter options are included in subsequent packets.

5. Protocol

By "initiator," we indicate the party that first includes filter options in its SYN packet. By "respondent," we indicate the other party.

The initiator includes a list of filters it is prepared to employ in its SYN packet. The respondent includes a (possibly empty) subset of those options in its first ACK packet. Both sides are then committed to applying precisely the filters indicated by the respondent to the payload in each subsequent packet. (Cf., Behavior.)

In the event of a simultaneous open, both sides will use the filters that form the intersection of the filter requests in the two SYN packets. This is confirmed by the two SYN+ACK packets.

Both initiator and respondent MUST NOT include payload data in any SYN packet that includes filter options. Similarly, the contacted party MUST NOT act as an initiator if the initial SYN packet includes payload data.

Because TCP options do not consume sequence number space, and hence are not acknowledged, each party MUST include precisely the options indicated by the respondent in all packets until it receives an ACK for a packet that contained at least one data byte, i.e., until it is guaranteed that the other party has acknowledged those options. Thereafter, the party does not include filter options in any subsequent packet.

6. Behavior

If the respondent has indicated that it can accept a filter, a sender MUST use it. If multiple filters apply to a packet (cf., Protocol), it indicates that all of those filters were applied to the packet. Filters are applied by the receiving system in the order in which the options were specified by the respondent (cf., Protocol). Thus, if the options following the base TCP header denote "ENCRYPTION" and "COMPRESSION", the receiving system must first decrypt the packet and then uncompress it. Obviously, the sender must apply the filters in the opposite order, first compressing the packet and then encrypting it.

Filters are strictly layered. There should not be interactions between different layers, though it is perfectly proper for one layer to assume the standard behavior provided by another. In particular, while it is perfectly proper for a filter to assume that TCP provides a reliable byte stream, it is not proper for it to assume any particular packet construction. If nothing else -- and there are many other reasons -- TCP's window size management, retransmission, and the coalescence of different segments into one during retransmission would wreak havoc on any filter that did make such assumptions.

6.1. Filter Selection

Again, because of the unpredictability of TCP's actual packetization, the filters to be used MUST be constant during a connection. The first packet sent after the initial SYN for each side MUST include the filter options that will be used during all transmissions for that connection. This specifically includes the ACK packet that finishes the three-way handshake.

Because filter options appear only so far as to acknowledge respondent's final choices, parameters in a given filter option are fixed by the SYN negotiation for the duration of a connection. They may be changed only by in-band communication, i.e., by the filter itself interpreting the payload data.

6.2. Header Compression

Packets with filter options may be subject to header compression [RFC1144]. There is no problem in doing so. Changes to the TCP options field can have a negative effect on header compression; this should not present a serious problem, however, since the options

occur only in the initial packets of a connection.

6.3. TCP URGENT Markings

Since TCP proper is unaware of the behavior (or even the existence) of any filter modules, it will simply note the URGENT pointer as usual. It is the responsibility of filter modules to handle this properly, including notifying the application (or any higher layers) as needed. Filters that change the length of the application-supplied data stream, such as compression modules, MUST make the corresponding adjustments to the effective URGENT pointer that is passed up.

7. Open Issues

The exact option syntax proposed here may not be ideal. To conserve option space, it may be useful to make all filters suboptions to a single TCP filter option.

It may also be useful to use separate options for request and transmission. That would avoid possible race conditions in the event of complex retransmissions of SYN+ACK packets or simultaneous opens, or to permit different filters to be used in each direction. In the simultaneous open case, separate options would also permit proper acquiescence to a filter request by a host that supports the option but had not requested it.

We considered allowing packet-level filter specification, by including appropriate filter options in each data-carrying packet. This would allow out-of-band modulation of the filter behavior during the connection. While such packet-level specification is theoretically possible, we anticipate that making the protocol robust, in particular vis-a-vis packet retransmission and coalescence, will be hard enough to make it impractical.

8. Security Considerations

Filtering data above the TCP layer should not have any negative impact on security. In particular, port numbers are not affected. Some firewalls and intrusion detection systems examine TCP payload data, however, and they may be confused by some filters. The former may wish to delete filter options; if the latter are used, administrators may wish to disable such options. Particular filter options, such as encryption, may have their own security considerations.

9. Acknowledgements

10. References

11. Author Information

Steven M. Bellovin
smb@research.att.com
+1 973-360-8656

Adam L. Buchsbaum
alb@research.att.com
+1 973-360-8674

S. Muthukrishnan
muthu@research.att.com
+1 973-360-7212

AT&T Labs--Research
Shannon Laboratory
180 Park Avenue
Florham Park, NJ 07974