US 20140283004A1

(19) **United States**

(12) **Patent Application Publication**

Moore

(10) **Pub. No.: US 2014/0283004 A1**

(43) **Pub. Date: Sep. 18, 2014**

(54) **FILTERING NETWORK DATA TRANSFERS**

(71) Applicant: **CENTRIPETAL NETWORKS, INC.**, Leesburg, VA (US)

(72) Inventor: **Sean Moore**, Hollis, NH (US)

(73) Assignee: **CENTRIPETAL NETWORKS, INC.**, Leesburg, VA (US)

(21) Appl. No.: **13/795,822**

(22) Filed: **Mar. 12, 2013**

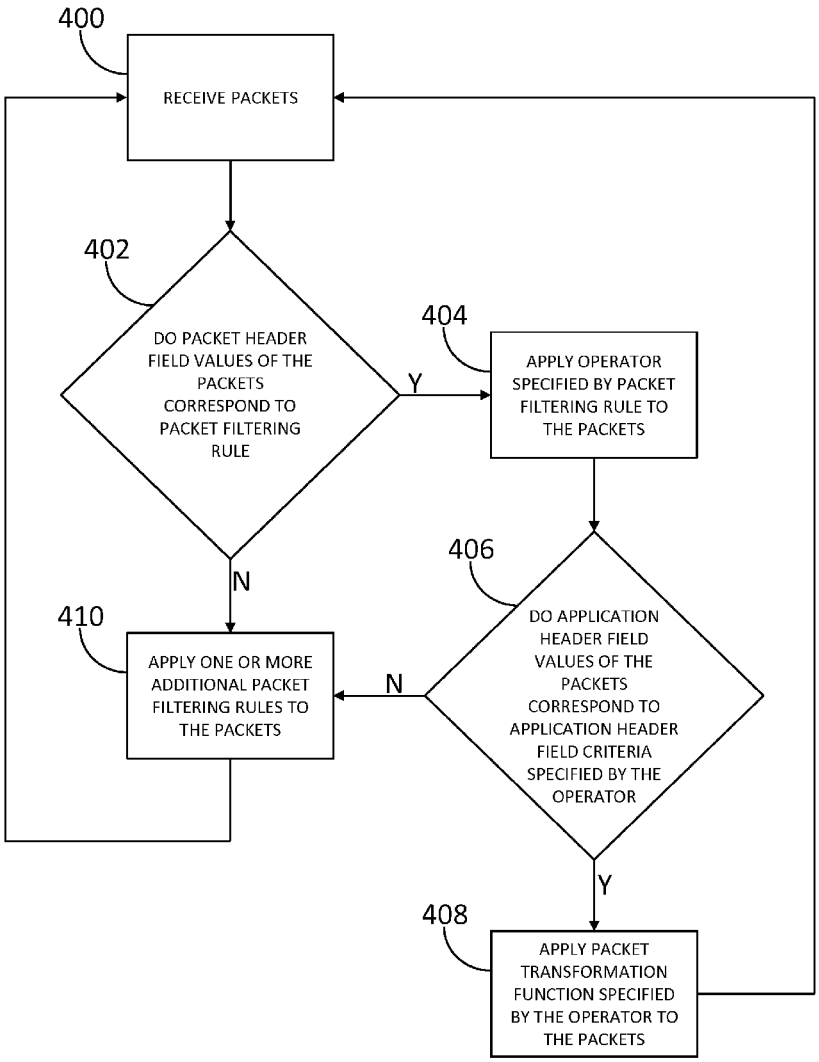
Publication Classification

(51) **Int. Cl.**
H04L 29/06 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 63/0263*** (2013.01)
USPC **726/13**

(57) **ABSTRACT**

Aspects of this disclosure relate to filtering network data transfers. In some variations, multiple packets may be received. A determination may be made that a portion of the packets have packet header field values corresponding to a packet filtering rule. Responsive to such a determination, an operator specified by the packet filtering rule may be applied to the portion of packets having the packet header field values corresponding to the packet filtering rule. A further determination may be made that one or more of the portion of the packets have one or more application header field values corresponding to one or more application header field criteria specified by the operator. Responsive to such a determination, at least one packet transformation function specified by the operator may be applied to the one or more of the portion of the packets.



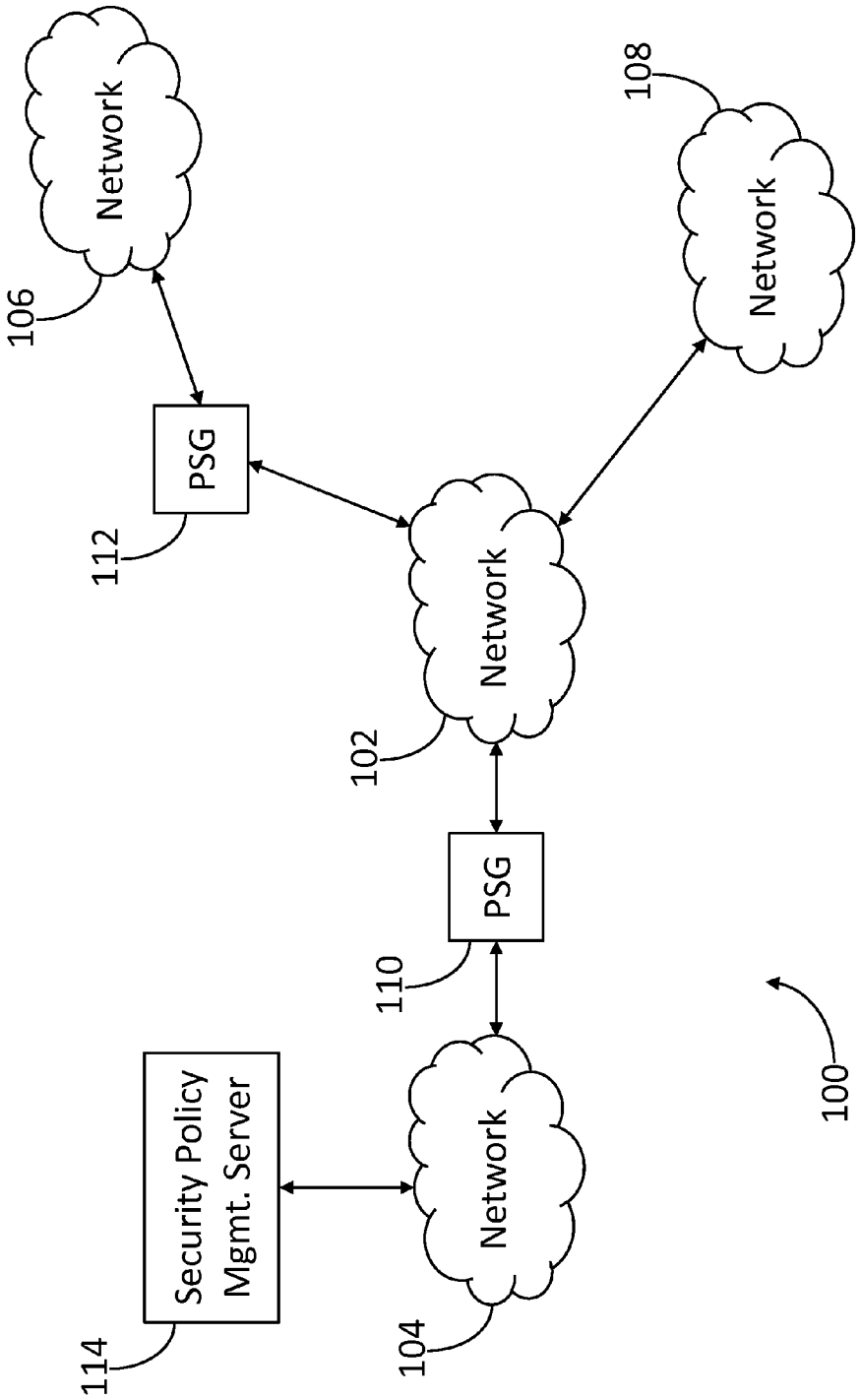

Network
106
PSG
112
Network
102
Network
108
PSG
110
Network
104
Security Policy Mgmt. Server
114
100

FIG. 1

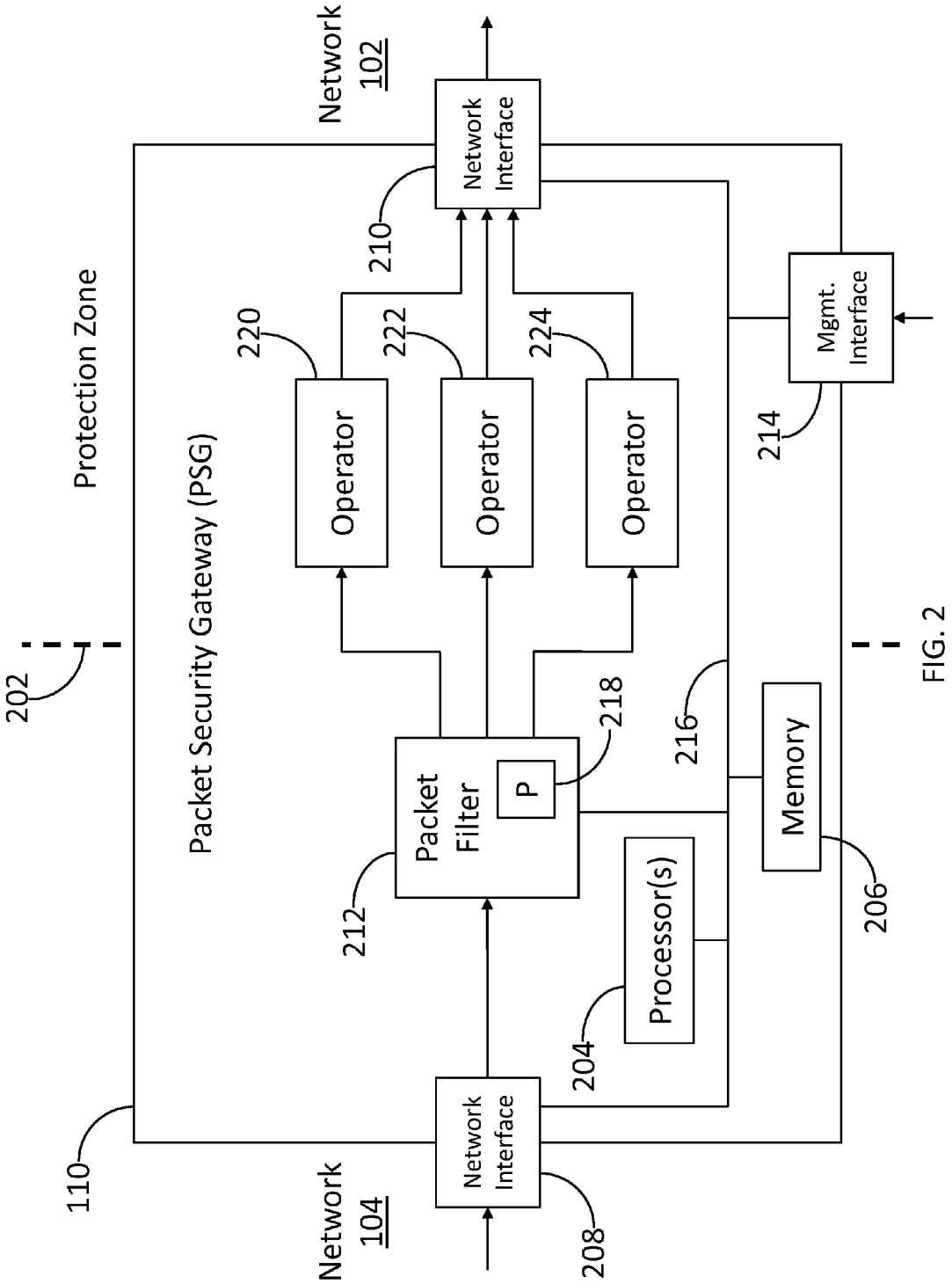
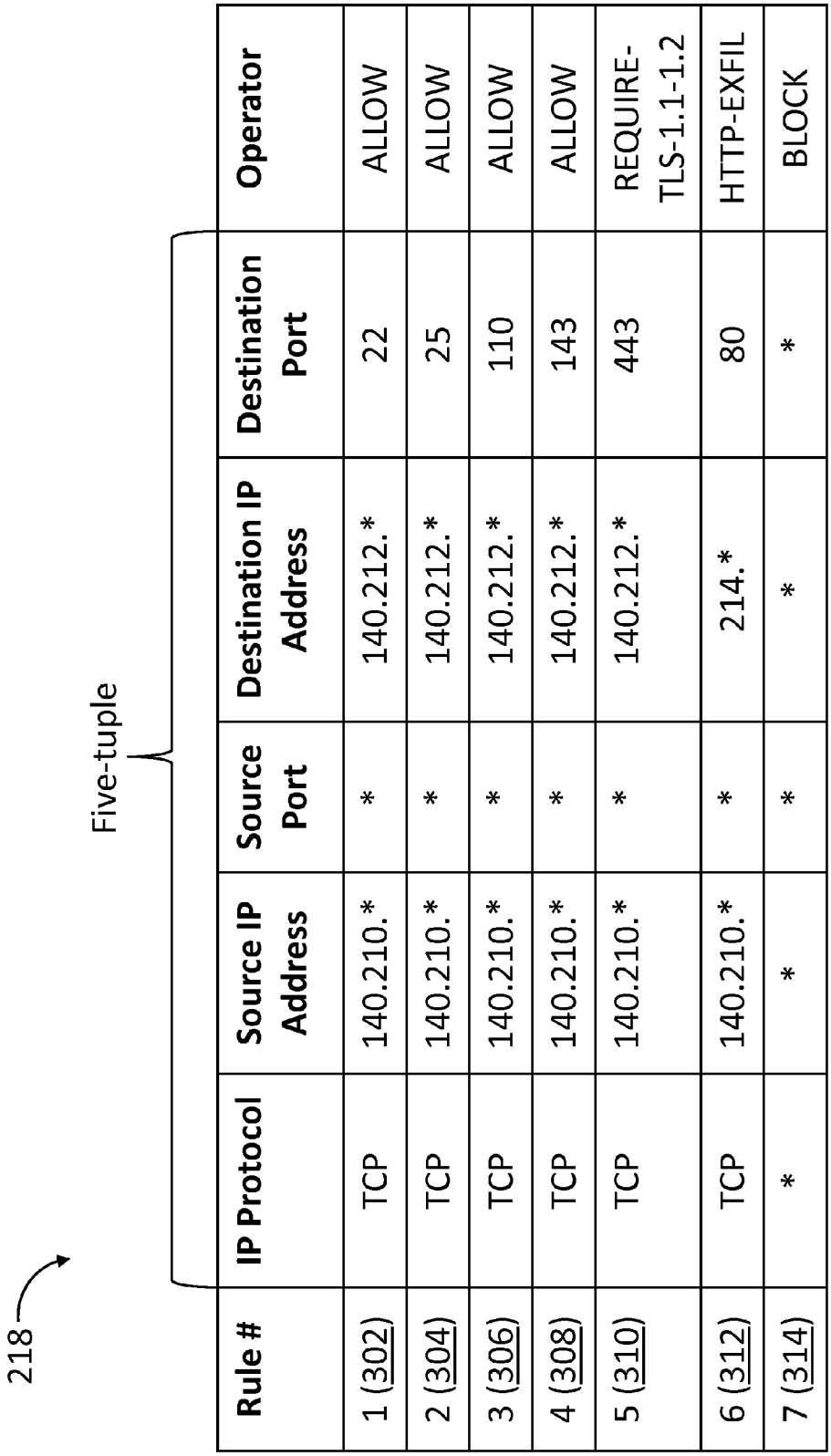

Network
102
Network Interface
210
220
222
224
Operator
Operator
Operator
Mgmt. Interface
214
Protection Zone
202
Packet Security Gateway (PSG)
Packet Filter
P
218
216
212
Memory
206
Processor(s)
204
Network Interface
208
Network
104
110

FIG. 2



Rule #	IP Protocol	Source IP Address	Source Port	Destination IP Address	Destination Port	Operator
1 (302)	TCP	140.210.*	*	140.212.*	22	ALLOW
2 (304)	TCP	140.210.*	*	140.212.*	25	ALLOW
3 (306)	TCP	140.210.*	*	140.212.*	110	ALLOW
4 (308)	TCP	140.210.*	*	140.212.*	143	ALLOW
5 (310)	TCP	140.210.*	*	140.212.*	443	REQUIRE-TLS-1.1-1.2
6 (312)	TCP	140.210.*	*	214.*	80	HTTP-EXFIL
7 (314)	*	*	*	*	*	BLOCK

FIG. 3

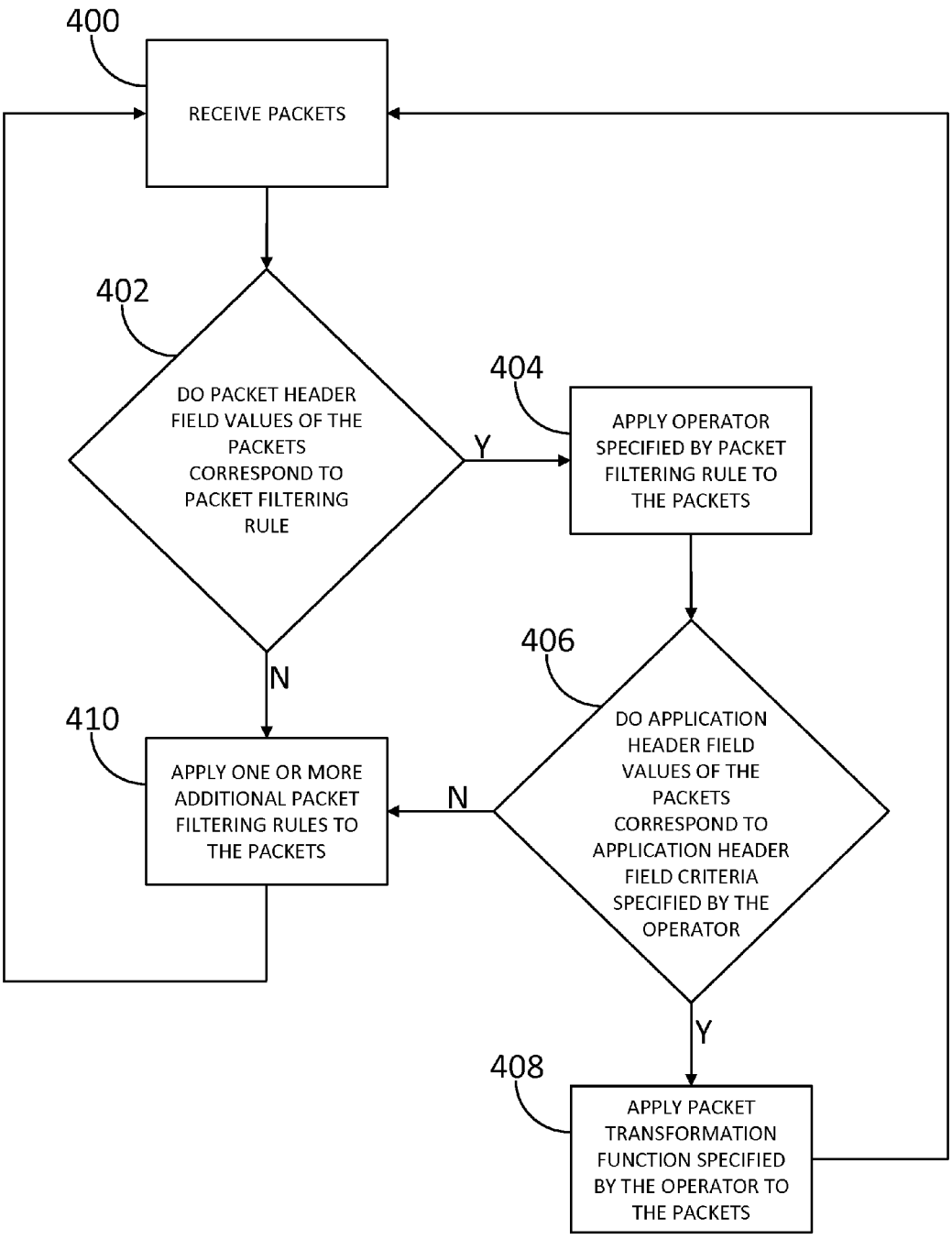

400
RECEIVE PACKETS
402
DO PACKET HEADER FIELD VALUES OF THE PACKETS CORRESPOND TO PACKET FILTERING RULE
Y
404
APPLY OPERATOR SPECIFIED BY PACKET FILTERING RULE TO THE PACKETS
N
406
DO APPLICATION HEADER FIELD VALUES OF THE PACKETS CORRESPOND TO APPLICATION HEADER FIELD CRITERIA SPECIFIED BY THE OPERATOR
N
410
APPLY ONE OR MORE ADDITIONAL PACKET FILTERING RULES TO THE PACKETS
Y
408
APPLY PACKET TRANSFORMATION FUNCTION SPECIFIED BY THE OPERATOR TO THE PACKETS

FIG. 4

FILTERING NETWORK DATA TRANSFERS

BACKGROUND

[0001] The TCP/IP network protocols (e.g., the Transmission Control Protocol (TCP) and the Internet Protocol (IP)) were designed to build large, resilient, reliable, and robust networks. Such protocols, however, were not originally designed with security in mind. Subsequent developments have extended such protocols to provide for secure communication between peers (e.g., Internet Protocol Security (IPsec)), but the networks themselves remain vulnerable to attack (e.g., Distributed Denial-of-Service (DDoS) attacks, phishing attacks, and the like).

[0002] A category of cyber attack known as exfiltrations (e.g., stealing sensitive data or credentials via the Internet) has proven to be especially difficult for conventional cyber defense systems to prevent. A first cause is that many exfiltrations are facilitated by using popular network data transfer protocols, such as the Hypertext Transfer Protocol (HTTP) used by the World Wide Web, that often appear to an observer (e.g., a conventional cyber defense system) as normal network behavior. A second cause is that typical network trust models, such as those used by network firewalls, interpret exfiltrations as trusted operations. A third cause is that human users often knowingly or unknowingly engage in network activities that are vulnerable to attack. A fourth cause is the general inability of conventional cyber defense systems to scale sufficiently to counter a cyber threat; for example, with respect to traffic volumes, network link speeds, network performance (e.g., latency and packet loss requirements), network usage policy enforcement, etc. Accordingly, many cyber attacks (e.g., DDoS attacks and exfiltrations) leverage Internet-scale characteristics to achieve their goals. Moreover, beyond those enumerated here, there are other causes for the failure of conventional, state-of-the-art cyber defense systems to prevent cyber attacks, such as exfiltrations.

SUMMARY

[0003] The following presents a simplified summary in order to provide a basic understanding of some aspects of the disclosure. It is neither intended to identify key or critical elements of the disclosure nor to delineate the scope of the disclosure. The following summary merely presents some concepts in a simplified form as a prelude to the detailed description below.

[0004] Aspects of this disclosure relate to filtering network data transfers. In some variations, multiple packets may be received. A determination may be made that a portion of the packets have packet header field values corresponding to a packet filtering rule. Responsive to such a determination, an operator specified by the packet filtering rule may be applied to the portion of packets having the packet header field values corresponding to the packet filtering rule. A further determination may be made that one or more of the portion of the packets have one or more application header field values corresponding to one or more application header field criteria specified by the operator. Responsive to such a determination, at least one packet transformation function specified by the operator may be applied to the one or more of the portion of the packets.

[0005] In some embodiments, a network packet filter may be located at a boundary between a secured network and an unsecured network (e.g., the Internet). The filter may observe packets traversing the network link between the secured network and the unsecured network. The filter may be capable of comparing certain packet header information (e.g., source and destination IP address(s), source and destination port(s), and protocol type(s)) with one or more packet filtering rules, which may define a network usage policy or network security policy. One or more of the rules may be associated with an operator that may be applied to a packet that matches one or more criteria specified by the rule.

[0006] Such packet filters may implement at least two operators: an identity operator, which may allow the packet to continue towards its destination, and a null operator which may prevent, or block, the packet from continuing towards its destination. In some embodiments, the network packet filter may implement one or more additional operators having the capability to determine if a packet contains an application-level header that specifies a particular method associated with a data transfer protocol; and, if so, whether to apply an identity operator or null operator to the packet. To distinguish a network packet filter that implements said additional operators from network packet filters that do not, a network packet filter that implements such additional operators will be referred to hereinafter as a packet security gateway (PSG).

[0007] For example, such an operator may be able to perform one or more of the following functions: (1) determine if an IP packet traversing a boundary contains an HTTP packet (e.g., an application-level HTTP packet) that specifies one or more specific HTTP methods (e.g., GET, PUT, POST, etc.), and (2) allow the packet (e.g., if a GET method is specified), or block the packet (e.g., if a PUT or POST method is specified). One or more administrators of the secured network may associate such an operator with one or more rules in a network security policy in order to enforce, via the PSG, a Web usage policy that may, for example, allow users to surf (e.g., GET) to one or more web sites attached to the Internet, but may prevent such user(s) from one or more of writing (e.g., PUT) data files or posting (e.g., POST) forms to one or more web sites. For example, administrator(s) may utilize such functionality to prevent one or more exfiltrations (e.g., file transfers containing sensitive information, posting of login credentials (usernames and passwords), etc.) to network nodes (e.g., web sites) that they may not trust.

[0008] Other details and features will be described in the sections that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present disclosure is pointed out with particularity in the appended claims. Features of the disclosure will become more apparent upon a review of this disclosure in its entirety, including the drawing figures provided herewith.

[0010] Some features herein are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings, in which like reference numerals refer to similar elements.

[0011] FIG. **1** illustrates an exemplary network environment in which one or more aspects of the disclosure may be implemented.

[0012] FIG. **2** illustrates an exemplary packet security gateway.

[0013] FIG. **3** illustrates an exemplary dynamic security policy with operators that filter on data transfer protocol, or application-layer protocol, header information.

[0014] FIG. **4** illustrates an exemplary method for protecting a secured network by enforcing one or more network usage policies or network security policies.

DETAILED DESCRIPTION

[0015] In the following description of various illustrative embodiments, reference is made to the accompanying drawings, which form a part hereof, and in which is shown, by way of illustration, various embodiments in which aspects of the disclosure may be practiced. It is to be understood that other embodiments may be utilized, and structural and functional modifications may be made, without departing from the scope of the present disclosure.

[0016] Various connections between elements are discussed in the following description. These connections are general and, unless specified otherwise, may be direct or indirect, wired or wireless, physical or logical. In this respect, the specification is not intended to be limiting.

[0017] FIG. **1** illustrates an exemplary network environment **100** in which one or more aspects of the disclosure may be implemented. Referring to FIG. **1**, network **102** may function as an interconnect between networks **104**, **106**, and **108**. For example, network **102** may be the public Internet, or some other large TCP/IP network functioning as an interconnect between one or more Local Area Networks (LANs) or Wide-Area Networks (WANs), (e.g., the Non-classified Internet Protocol (IP) Router Network (NIPRNet), operated by the United States Defense Information Systems Agency (DISA)). Networks **104**, **106**, and **108** may be LANs or WANs operated by or otherwise associated with various organizations (e.g., one or more commercial enterprises, companies, universities, military commands, government agencies, or cyber criminal organizations).

[0018] For example, a geographically distributed commercial enterprise X may own and operate networks **104** and **106**, and use network **102** (e.g., the Internet) to interconnect networks **104** and **106**, and to access other networks (e.g., other networks not owned or operated by enterprise X) attached to network **102**. One or more computing devices (e.g., workstations, servers, etc.) of enterprise X may be attached to network **104** or **106**. Network **108** may be owned and operated by a cyber criminal organization Z, which may attempt to steal information (e.g., sensitive data) from enterprise X, for example, via network **102**. Members of organization Z may attach one or more computing devices (e.g., workstations or servers) to network **108**, and may use these workstation(s) or server(s) to attack or collect data from one or more networks affiliated with enterprise X (e.g., network **104** or **106**).

[0019] As used herein, a packet security gateway (PSG) may include one or more computing devices configured to receive packets, and apply one or more filters or operators, including an identity (e.g., allow) or null (e.g., block) operator, to the packets. In some embodiments, a packet security gateway may be configured to apply one or more additional operators as described herein. As used herein, a security policy management server may include one or more computing devices configured to communicate a dynamic security policy to a packet security gateway. In some embodiments, a security policy management server may be configured to perform one or more additional functions as described herein. As used herein, a dynamic security policy may include one or more rules, messages, instructions, files, data structures, or the like specifying criteria corresponding to one or more packets and may identify one or more operators to be applied to packets corresponding to the specified criteria. In some embodiments, a dynamic security policy may specify one or more additional parameters as described herein.

[0020] Network environment **100** may include one or more packet security gateways and one or more security policy management servers. For example, network environment **100** may include packet security gateways **110** and **112**, and security policy management server **114**. One or more security policy management servers may be associated with a protected network. For example, networks **104** and **106** may each be distinct LANs associated with a common enterprise X, and may each form part of a protected or secured network associated with security policy management server **114**. Enterprise X may desire to prevent cyber attacks (e.g., exfiltrations) from one or more of its networks (e.g., network **104** or **106**). Accordingly, it may locate one or more packet security gateways at each boundary between its networks and one or more public interconnect networks (e.g., network **102**), which may be utilized by a cyber criminal, such as organization Z, to attempt to remotely access its networks (e.g., network **104** or **106**), and which may, for example, potentially be used to attempt to transfer data from one or more of its networks (e.g., network **104** or **106**) to one or more networks affiliated with organization Z (e.g., network **108**). For example, packet security gateway **110** may protect network **104** from one or more cyber attacks (e.g., exfiltrations) mediated by network **102** (e.g., the Internet), and packet security gateway **112** may protect network **106** from one or more cyber attacks (e.g., exfiltrations) mediated by network **102**.

[0021] As will be described in greater detail below, each of one or more packet security gateways associated with a security policy management server may be configured to receive a dynamic security policy from a security policy management server, receive packets associated with a network protected by the packet security gateway, and perform one or more operations specified by the dynamic security policy on the packets. For example, each of packet security gateways **110** and **112** may be configured to receive a dynamic security policy from security policy management server **114**. Each of packet security gateways **110** and **112** may also be configured to receive packets associated with networks **104**, **106**, or **108**. Each of packet security gateways **110** and **112** may further be configured to apply one or more rules or operators specified by the dynamic security policy received from security policy management server **114** to packets associated with networks **104**, **106**, or **108**.

[0022] FIG. **2** illustrates an exemplary packet security gateway according to one or more aspects of the disclosure. Referring to FIG. **2**, as indicated above, packet security gateway **110** may be located at network boundary **202** between networks **104** and **102**. Packet security gateway **110** may include one or more processors **204**, memory **206**, network interfaces **208** and **210**, packet filter **212**, and management interface **214**. Processor(s) **204**, memory **206**, network interfaces **208** and **210**, packet filter **212**, and management interface **214** may be interconnected via data bus **216**. Network interface **208** may connect packet security gateway **110** to network **104**. Similarly, network interface **210** may connect packet security gateway **110** to network **102**. Memory **206** may include one or more program modules that when executed by processor(s) **204**, may configure packet security gateway **110** to perform one or more of various functions described herein.

[0023] Packet security gateway **110** may be configured to receive a dynamic security policy from security policy man-

agement server **114**. For example, packet security gateway **110** may receive dynamic security policy **218** from security policy management server **114** via management interface **214** (e.g., via out-of-band signaling) or network interface **208** (e.g., via in-band signaling). Packet security gateway **110** may include one or more packet filters or packet discriminators, or logic for implementing one or more packet filters or packet discriminators. For example, packet security gateway **110** may include packet filter **212**, which may be configured to examine information associated with packets received by packet security gateway **110** and forward such packets to one or more of operators **220**, **222**, or **224** based on the examined information. For example, packet filter **212** may examine information associated with packets received by packet security gateway **110** (e.g., packets received from network **104** via network interface **208**) and forward the packets to one or more of operators **220**, **222**, or **224** based on the examined information.

[0024] As will be described in greater detail below, dynamic security policy **218** may include one or more rules and the configuration of packet filter **212** may be based on one or more of the rules included in dynamic security policy **218**. For example, dynamic security policy **218** may include one or more rules specifying that packets having specified information should be forwarded to operator **220**, that packets having different specified information should be forwarded to operator **222**, and that all other packets should be forwarded to operator **224**. Operators **220**, **222**, and **224** may be configured to perform one or more functions on packets they receive from packet filter **212**. For example, one or more of operators **220**, **222**, or **224** may be configured to forward packets received from packet filter **212** into network **102**, forward packets received from packet filter **212** to an IPsec stack (not illustrated) having an IPsec security association corresponding to the packets, or drop packets received from packet filter **212**. In some embodiments, one or more of operators **220**, **222**, or **224** may be configured to drop packets by sending the packets to a local "infinite sink" (e.g., the /dev/null device file in a UNIX/LINUX system).

[0025] FIG. **3** illustrates an exemplary dynamic security policy in accordance with one or more embodiments. Referring to FIG. **3**, dynamic security policy **218** may include rules 1 **302**, 2 **304**, 3 **306**, 4 **308**, 5 **310**, 6 **312**, and 7 **314**. Each of these rules may specify criteria and one or more operators that may be applied to packets associated with (e.g., matching) the specified criteria. The specified criteria may take the form of a five-tuple, which may, for example, comprise one or more values selected from, packet header information, specifying a protocol type of the data section of an IP packet (e.g., TCP, User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), or one or more other protocols), one or more source IP addresses, one or more source port values, one or more destination IP addresses, and one or more destination ports.

[0026] For example, rule 1 **302** may specify that IP packets containing one or more TCP packets, originating from a source IP address that begins with 140.210, having any source port, destined for an IP address that begins with 140.212, and destined for port 22 (e.g., associated with the Secure Shell (SSH) protocol) should have an ALLOW operator (e.g., an identity operator) applied to them. Similarly, rule 2 **304** may specify that IP packets containing one or more TCP packets, originating from a source IP address that begins with 140.210, having any source port, destined for an IP address that begins with 140.212, and destined for port 25 (e.g., associated with the Simple Mail Transfer Protocol (SMTP)) should have an ALLOW operator applied to them.

[0027] Rule 3 **306** may specify that IP packets containing one or more TCP packets, originating from a source IP address that begins with 140.210, having any source port, destined for an IP address that begins with 140.212, and destined for port 110 (e.g., associated with Post Office Protocol version 3 (POP3)) should have an ALLOW operator applied to them.

[0028] Rule 4 **308** may specify that IP packets containing one or more TCP packets, originating from a source IP address that begins with 140.210, having any source port, destined for an IP address that begins with 140.212, and destined for port 143 (e.g., associated with the Internet Message Access Protocol (IMAP)) should have an ALLOW operator applied to them.

[0029] Rule 5 **310** may specify that IP packets containing one or more TCP packets, originating from a source IP address that begins with 140.210, having any source port, destined for an IP address that begins with 140.212, and destined for port 443 (e.g., associated with the port for the Hypertext Transfer Protocol Secure (HTTPS) protocol) should have a specified Transport Layer Security (TLS) protocol (e.g., REQUIRE-TLS 1.1-1.2) operator (as described in greater detail below) applied to them.

[0030] Rule 7 **314** may be a "wildcard" rule and may apply a BLOCK operator (e.g., a null operator which "drops" any packets it is applied to) to any packets that do not match the criteria of any of Rules 1 **302**, 2 **304**, 3 **306**, 4 **308**, 5 **310**, or 6 **312** (e.g., when rules 1 **302**, 2 **304**, 3 **306**, 4 **308**, 5 **310**, 6 **312**, and 7 **314** are applied to packets in a linear fashion).

[0031] As described above with respect to FIG. **1**, networks **104** and **106** may be owned or operated by enterprise X. Enterprise X may have allocated IPv4 addresses 140.210.0.0/16 to network **104**, and IPv4 addresses 140.212.0.0/16 to network **106**. Enterprise X may have loaded PSG **110** with dynamic security policy **218**, and may utilize PSG **110** to enforce one or more network security policies embodied in one or more rules of dynamic security policy **218** to restrict network communications between networks **104** and **106** (e.g., to secure system logins, e-mail, encrypted web sessions, and the like). For example, based on standard usage of ports, rule 1 **302** may allow any hosts attached to network **104** to conduct Secure Shell (SSH) sessions (e.g., system logins) with any hosts attached to network **106**; rule 2 **304** may allow any e-mail servers attached to network **104** to conduct SMTP sessions (e.g., e-mail transfer sessions) with any e-mail servers attached to network **106**; rule 3 **306** and rule 4 **308**, may respectively allow e-mail clients attached to network **104** to conduct POP3 and IMAP sessions (e.g., e-mail download sessions into a webmail browser application) with any e-mail servers attached to network **106**; and rule 5 **310** may allow web browsers attached to network **104** to conduct HTTPS sessions (e.g., secure web sessions) with any web servers attached to network **106**, but may, as described in greater detail below, utilize the REQUIRE-TLS-1.1-1.2 operator to ensure that only HTTPS secure web sessions using version 1.1 or 1.2 of the Transport Layer Security (TLS) protocol to secure such HTTPS sessions are allowed (e.g., because the popular TLS version 1.0 protocol has a known security vulnerability that attackers may exploit to decrypt HTTPS sessions).

[0032] Rule 6 **312** may specify that IP packets containing one or more TCP packets, originating from a source IP address that begins with 140.210., having any source port, destined for an IP address that begins with 200.214, and destined for port 80 (e.g., associated with the HTTP protocol) should have an HTTP-EXFIL operator applied to them. As described in greater detail below, an HTTP-EXFIL operator may allow HTTP packets containing a GET method, but may block HTTP packets containing other HTTP methods (e.g., PUT, POST, CONNECT, etc.). Such an operator may thus allow a web browser to "surf the web" (e.g., download web pages hosted by web servers), but may prevent the web browser from writing files to a web server (e.g., using the PUT method), posting forms (e.g., forms that might contain login credentials, such as usernames or passwords) to a web server (e.g., using the POST method), or otherwise communicating with a web server (e.g., using any HTTP method except GET). Because attackers may often use HTTP PUT or POST methods to exfiltrate sensitive data, operators such as HTTP-EXFIL may be used to stop such exfiltrations.

[0033] Returning to the example described above, organization Z may own or operate network **108**, and may have allocated network IP addresses 214.0.0.0/8 to network **108**. Enterprise X may not have a business relationship with organization Z, and may therefore not consider network **108** to be trusted. While enterprise X could simply block all communications to networks that are owned or operated by organizations it does not fully trust, this would likely result in enterprise X blocking access to most of the Internet. Enterprise X's employees, therefore, could not freely surf the Web, which may restrict the employees' ability to conduct business on behalf of enterprise X. By enforcing sets of rules similar to rule 6 **312** that may apply operators like or similar to HTTP-EXFIL, enterprise X may enable its employees to freely surf the web and conduct company business, but may prevent one or more cyber attacks (e.g., HTTP-mediated exfiltrations).

[0034] One function of operators like HTTP-EXFIL and REQUIRE-TLS-1.1-1.2 may be to inspect fields in the headers of application packets contained in IP packets, determine field values, and depending on the field values, decide to allow, block, or otherwise apply a packet transformation function (e.g., encapsulate the packet into a tunnel, alter one or more header field values, etc.) to the packets. The logic for this function may be executed by one or more of operators **220**, **222**, or **224**. The logic may be developed in a high-level programming language such as C. An example of such programmatic logic, written in pseudocode, for the HTTP-EXFIL operator, is as follows:

```
Operator HTTP-EXFIL(ip-packet):
        Inspect app-pkt(ip-packet) match GET return ALLOW;
        Inspect app-pkt(ip-packet) match POST return BLOCK;
        Inspect app-pkt(ip-packet) match PUT return BLOCK;
        Inspect app-pkt(ip-packet) match DELETE return BLOCK;
        Inspect app-pkt(ip-packet) match CONNECT return BLOCK;
        Return BLOCK;
End Operator HTTP-EXFIL;
```

[0035] Referring to the above-example, Operator HTTP-EXFIL may accept as input an IP packet that matches a rule, such as rule 6 **312** of dynamic security policy **218**. If the application packet contained in the IP packet is an HTTP packet, the value of the HTTP method field in the HTTP packet header may be compared to the values that encode the GET, POST, PUT, DELETE, and CONNECT methods. If a match is found, then the HTTP-EXFIL operator may return either ALLOW or BLOCK, depending on the method value. If no match is found, then the HTTP-EXFIL operator may return BLOCK.

[0036] An example of programmatic logic, written in pseudocode, for a REQUIRE-TLS-1.1-1.2 operator is provided below. The REQUIRE-TLS-1.1-1.2 operator may be associated with filter rules for HTTPS sessions, such as rule 5 **310** of dynamic security policy **218**. HTTPS may be used to encrypt HTTP sessions. HTTPS is not a protocol per se, but rather the result of layering the HTTP protocol on top of the TLS protocol. For an HTTPS session composed of IP packets, the application packets contained in the IP packets may be TLS Record Protocol packets. The header fields of TLS Record Protocol packets may not be encrypted. One of the header fields may contain a value indicating the TLS version.

[0037] Exemplary programmatic logic for a REQUIRE-TLS-1.1-1.2 operator, written in pseudocode, is as follows:

```
Operator REQUIRE-TLS-1.1-1.2(ip-packet):
        Inspect app-pkt(ip-packet) match 1.0 return BLOCK;
        Inspect app-pkt(ip-packet) match 1.1 return ALLOW;
        Inspect app-pkt(ip-packet) match 1.2 return ALLOW;
        Return BLOCK;
End Operator REQUIRE-TLS-1.1-1.2;
```

[0038] Referring to the above-example, Operator REQUIRE-TLS-1.1-1.2 may accept as input an IP packet that matches a rule, such as rule 5 **310** of dynamic security policy **218**. If the application packet contained in the IP packet is a TLS Record Protocol packet, the value of the version field in the TLS Record Protocol packet header may be compared to the values that encode version numbers 1.0, 1.1, and 1.2. If a match is found, then the REQUIRE-TLS-1.1-1.2 operator may return either ALLOW or BLOCK, depending on the version number value. If no match is found, then the REQUIRE-TLS-1.1-1.2 operator may return BLOCK.

[0039] The filtering process described herein may be viewed as having two (2) stages: A first stage in which the "5-tuple" of IP packet header field values and transport protocol (e.g., TCP, UDP, etc.) packet header field values may be filtered; and, a second stage in which application packet header field values may be filtered (e.g., by applying operator logic similar to that described above). Conceptually, the first stage may determine if the network policy allows any communications between the resources identified in the 5-tuple rule; if so, the second stage may determine if the policy allows the specific method or type of communication (e.g., file read, file write, encrypted communication, etc.) between the resources. Such a method may, however, be used in other conceptual models.

[0040] The methods described above may be modified to achieve different functionality and may be extended to other data transfer protocols or to other application-layer protocols. These methods may provide network administrators with capabilities to enforce network usage policies and network security policies that have capabilities and functionalities beyond those described above. For example, these methods may provide network administrators with capabilities to prevent exfiltrations that are mediated by other data transfer protocols besides HTTP and HTTPS. Examples of such protocols include File Transfer Protocol (FTP) and messaging

protocols such as eXtensible Messaging and Presence Protocol (XMPP). Moreover, new network applications may emerge in the future which may use new data transfer protocols or application-layer protocols to which the present methods may be applied. These methods may also be used for purposes other than network policy enforcement and exfiltration prevention. For example, it may be useful for a packet filter to rapidly detect if an IP packet contains a Real-time Transport Protocol (RTP) application packet used to deliver audio or video information (e.g., if a cyber attack based on RTP has yet to be discovered, network administrators may choose to not process RTP packets through the cyber security defense systems that may be protecting their networks).

[0041] FIG. **4** illustrates an exemplary method for protecting a secured network in accordance with one or more embodiments of the present disclosure. The steps may be performed at one or more packet security gateways associated with a security policy management server. For example, each of packet security gateways **110** and **112** may be associated with security policy management server **114**, and the steps may be performed at packet security gateway **110** or **112**. At step **400**, packets may be received. For example, packet security gateway **110** may receive packets from network **104** via network interface **208** that are destined for network **106**. At step **402**, a determination may be made as to whether a portion of the received packets have packet header field values corresponding to a packet filtering rule. For example, a determination may be made as to whether a portion of the packets received from network **104** have packet header field values (e.g., one or more of one or more data section protocols, one or more source IP addresses, one or more source ports, one or more destination IP addresses, or one or more destination ports) corresponding to rule 5 **310**. At step **404**, responsive to determining that one or more of the portion of received packets have packet header field values corresponding to the packet filtering rule, an operator specified by the packet filtering rule may be applied to the portion of the received packets. For example, the REQUIRE TLS-1.1-1.2 operator specified by rule 5 **310** may be applied to the portion of the received packets.

[0042] At step **406**, a determination may be made as to whether one or more application header field values of one or more of the portion of the received packets correspond to one or more application header field criteria specified by the operator. For example, a determination may be made as to whether one or more of the portion of the received packets have application header field values corresponding to one or more application header field criteria of the REQUIRE TLS-1.1-1.2 operator specified by rule 5 **310** (e.g., application header field values corresponding to TLS version 1.1 or 1.2). At step **408**, responsive to determining that one or more of the portion of received packets have application header field values corresponding to one or more application header field criteria specified by the operator, a packet transformation function specified by the operator may be applied to the one or more of the portion of the received packets. For example, an ALLOW packet transformation function specified by the REQUIRE TLS-1.1-1.2 operator may be applied to the one or more of the portion of the received packets having application header field values corresponding to one or more application header field criteria of the REQUIRE TLS-1.1-1.2 operator specified by rule 5 **310** (e.g., each of the one or more of the portion of the received packets may be allowed to continue toward their respective destinations). The method may then return to step **400** and await receipt of one or more additional packets (e.g., one or more additional packets from network **104** received via network interface **208** that are destined for network **106**).

[0043] Returning to step **406**, a determination may be made as to whether one or more application header field values of one or more of the portion of the received packets correspond to one or more application header field criteria specified by the operator. For example, a determination may be made as to whether one or more of the portion of the received packets have application header field values corresponding to one or more application header field criteria of the REQUIRE TLS-1.1-1.2 operator specified by rule 5 **310** (e.g., application header field values corresponding to TLS version 1.1 or 1.2). Responsive to determining that one or more of the portion of received packets have application header field values that do not correspond to one or more application header field criteria specified by the operator, one or more additional packet filtering rules may be applied to the one or more of the portion of the received packets. For example, rule 7 **314** may be applied to the one or more of the portion of the received packets having application header field values that do not correspond to one or more application header field criteria of the REQUIRE TLS-1.1-1.2 operator specified by rule 5 **310** (e.g., each of the one or more of the portion of the received packets may be blocked from continuing toward their respective destinations). The method may then return to step **400** and await receipt of one or more additional packets (e.g., one or more additional packets from network **104** received via network interface **208** that are destined for network **106**).

[0044] Returning to step **402**, a determination may be made as to whether a portion of the received packets have packet header field values corresponding to a packet-filtering rule. For example, a determination may be made as to whether a portion of the packets received from network **104** have packet header field values (e.g., one or more of one or more data section protocols, one or more source IP addresses, one or more source ports, one or more destination IP addresses, or one or more destination ports) corresponding to rule 5 **310**. Responsive to determining that the portion of received packets have packet header field values that do not correspond to the packet filtering rule, one or more additional packet filtering rules may be applied to the one or more of the portion of the received packets. For example, rule 7 **314** may be applied to the portion of received packets that do not have packet header field values that correspond to rule 5 **310** (e.g., each of the portion of the received packets may be blocked from continuing toward their respective destinations). The method may then return to step **400** and await receipt of one or more additional packets (e.g., one or more additional packets from network **104** received via network interface **208** that are destined for network **106**).

[0045] The functions and steps described herein may be embodied in computer-usable data or computer-executable instructions, such as in one or more program modules, executed by one or more computers or other devices to perform one or more functions described herein. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types when executed by one or more processors in a computer or other data processing device. The computer-executable instructions may be stored on a computer-readable medium such as a hard disk, optical disk, removable storage media, solid-state memory, RAM,

etc. As will be appreciated, the functionality of the program modules may be combined or distributed as desired in various embodiments. In addition, the functionality may be embodied in whole or in part in firmware or hardware equivalents, such as integrated circuits, application-specific integrated circuits (ASICs), field programmable gate arrays (FPGA), and the like. Particular data structures may be used to more effectively implement one or more aspects of the disclosure, and such data structures are contemplated to be within the scope of computer executable instructions and computer-usable data described herein.

[0046] Although not required, one of ordinary skill in the art will appreciate that various aspects described herein may be embodied as a method, an apparatus, or as one or more computer-readable media storing computer-executable instructions. Accordingly, those aspects may take the form of an entirely hardware embodiment, an entirely software embodiment, an entirely firmware embodiment, or an embodiment combining software, hardware, and firmware aspects in any combination.

[0047] As described herein, the various methods and acts may be operative across one or more computing devices and one or more networks. The functionality may be distributed in any manner, or may be located in a single computing device (e.g., a server, a client computer, etc.).

[0048] Aspects of the disclosure have been described in terms of illustrative embodiments thereof. Numerous other embodiments, modifications, and variations within the scope and spirit of the appended claims will occur to persons of ordinary skill in the art from a review of this disclosure. For example, one of ordinary skill in the art will appreciate that the steps illustrated in the illustrative figures may be performed in other than the recited order, and that one or more steps illustrated may be optional.

What is claimed is:

1. A method, comprising:

at a computing platform comprising at least one processor, memory, and communication interface:

receiving, via the communication interface, a plurality of packets;

determining, by the processor, that at least a portion of the plurality of packets have packet header field values corresponding to a packet filtering rule stored in the memory;

responsive to determining that the at least a portion of the plurality of packets have packet header field values corresponding to the packet filtering rule stored in the memory, applying an operator specified by the packet filtering rule to the at least a portion of the plurality of packets, wherein the operator specifies one or more application header field value criteria;

determining, by the processor, that one or more of the at least a portion of the plurality of packets have one or more application header field values corresponding to the one or more application header field value criteria specified by the operator; and

responsive to determining that the one or more of the at least a portion of the plurality of packets have one or more application header field values corresponding to the one or more application header field criteria specified by the operator, applying at least one packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets.

2. The method of claim **1**, wherein the at least a portion of the plurality of packets comprise Hypertext Transfer Protocol Secure (HTTPS) packets, and wherein the one or more application header field value criteria comprise one or more Transport Layer Security (TLS) version values.

3. The method of claim **2**, wherein the one or more TLS version values specify one or more TLS versions for which packets should be allowed to continue toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the TLS versions, and wherein applying the packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets comprises allowing each of the one or more of the at least a portion of the plurality of packets to continue toward their respective destinations.

4. The method of claim **2**, wherein the one or more TLS version values specify one or more TLS versions for which packets should be blocked from continuing toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the TLS versions, and wherein applying the packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets comprises blocking each of the one or more of the at least a portion of the plurality of packets from continuing toward their respective destinations.

5. The method of claim **1**, wherein the at least a portion of the plurality of packets comprise Hypertext Transfer Protocol (HTTP) packets, and wherein the one or more application header field value criteria comprise one or more HTTP method values.

6. The method of claim **5**, wherein the one or more HTTP method values specify one or more HTTP methods for which packets should be allowed to continue toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the HTTP methods, and wherein applying the packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets comprises allowing each of the one or more of the at least a portion of the plurality of packets to continue toward their respective destinations.

7. The method of claim **5**, wherein the one or more HTTP method values specify one or more HTTP methods for which packets should be blocked from continuing toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the HTTP methods, and wherein applying the packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets comprises blocking each of the one or more of the at least a portion of the plurality of packets from continuing toward their respective destinations.

8. The method of claim **5**, wherein the one or more of the at least a portion of the plurality of packets comprise a first portion of HTTP packets each of which comprises at least one GET method value and a second portion of HTTP packets each of which comprises one or more of PUT, POST, or CONNECT method values, and wherein applying the at least

one packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets comprises:

allowing each of the first portion of HTTP packets to continue toward their respective destinations; and

blocking each of the second portion of HTTP packets from continuing toward their respective destinations.

9. The method of claim **1**, wherein determining that the at least a portion of the plurality of packets have packet header field values corresponding to the packet filtering rule stored in the memory comprises:

comparing a protocol of a data section of each of the at least a portion of the plurality of packets to a protocol specified by the packet filtering rule;

comparing a source Internet Protocol (IP) address of each of the at least a portion of the plurality of packets to one or more source IP addresses specified by the packet filtering rule;

comparing a source port of each of the at least a portion of the plurality of packets to one or more source ports specified by the packet filtering rule;

comparing a destination IP address of each of the at least a portion of the plurality of packets to one or more destination IP addresses specified by the packet filtering rule; and

comparing a destination port of each of the at least a portion of the plurality of packets to one or more destination ports specified by the packet filtering rule.

10. The method of claim **1**, comprising:

receiving, via the communication interface, a second plurality of packets;

determining, by the processor, that the second plurality of packets do not have packet header field values corresponding to the packet filtering rule stored in the memory; and

responsive to determining that the second plurality of packets do not have packet header field values corresponding to the packet filtering rule, blocking each of the second plurality of packets from continuing toward their respective destinations without applying the operator specified by the packet filtering rule to the second plurality of packets.

11. An apparatus, comprising:

at least one processor; and

a memory storing instructions that when executed by the at least one processor cause the apparatus to:

receive a plurality of packets;

determine that at least a portion of the plurality of packets have packet header field values corresponding to a packet filtering rule;

responsive to determining that the at least a portion of the plurality of packets have packet header field values corresponding to the packet filtering rule, apply an operator specified by the packet filtering rule to the at least a portion of the plurality of packets, wherein the operator specifies one or more application header field value criteria;

determine that one or more of the at least a portion of the plurality of packets have one or more application header field values corresponding to the one or more application header field value criteria specified by the operator; and

responsive to determining that the one or more of the at least a portion of the plurality of packets have one or more application header field values corresponding to the one or more application header field criteria specified by the operator, apply at least one packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets.

12. The apparatus of claim **11**, wherein the at least a portion of the plurality of packets comprise Hypertext Transfer Protocol Secure (HTTPS) packets, and wherein the one or more application header field value criteria comprise one or more Transport Layer Security (TLS) version values.

13. The apparatus of claim **12**, wherein the one or more TLS version values specify one or more TLS versions for which packets should be allowed to continue toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the TLS versions, and wherein the instructions when executed by the at least one processor cause the apparatus to allow each of the one or more of the at least a portion of the plurality of packets to continue toward their respective destinations.

14. The apparatus of claim **12**, wherein the one or more TLS version values specify one or more TLS versions for which packets should be blocked from continuing toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the TLS versions, and wherein the instructions when executed by the at least one processor cause the apparatus to block each of the one or more of the at least a portion of the plurality of packets from continuing toward their respective destinations.

15. The apparatus of claim **11**, wherein the at least a portion of the plurality of packets comprise Hypertext Transfer Protocol (HTTP) packets, and wherein the one or more application header field value criteria comprise one or more HTTP method values.

16. The apparatus of claim **15**, wherein the one or more HTTP method values specify one or more HTTP methods for which packets should be allowed to continue toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the HTTP methods, and wherein the instructions when executed by the at least one processor cause the apparatus to allow each of the one or more of the at least a portion of the plurality of packets to continue toward their respective destinations.

17. The apparatus of claim **15**, wherein the one or more HTTP method values specify one or more HTTP methods for which packets should be blocked from continuing toward their respective destinations, wherein the one or more application header field values corresponding to the one or more application header field criteria specified by the operator comprise one or more of the HTTP methods, and wherein the instructions when executed by the at least one processor cause the apparatus to block each of the one or more of the at least a portion of the plurality of packets from continuing toward their respective destinations.

18. The apparatus of claim **15**, wherein the one or more of the at least a portion of the plurality of packets comprise a first portion of HTTP packets each of which comprises at least one GET method value and a second portion of HTTP packets

each of which comprises one or more of PUT, POST, or CONNECT method values, and wherein the instructions when executed by the at least one processor cause the apparatus to:

allow each of the first portion of HTTP packets to continue toward their respective destinations; and

block each of the second portion of HTTP packets from continuing toward their respective destinations.

19. The apparatus of claim **11**, wherein the instructions when executed by the at least one processor cause the apparatus to:

compare a protocol of a data section of each of the at least a portion of the plurality of packets to a protocol specified by the packet filtering rule;

compare a source Internet Protocol (IP) address of each of the at least a portion of the plurality of packets to one or more source IP addresses specified by the packet filtering rule;

compare a source port of each of the at least a portion of the plurality of packets to one or more source ports specified by the packet filtering rule;

compare a destination IP address of each of the at least a portion of the plurality of packets to one or more destination IP addresses specified by the packet filtering rule; and

compare a destination port of each of the at least a portion of the plurality of packets to one or more destination ports specified by the packet filtering rule.

20. One or more non-transitory computer-readable media having instructions stored thereon, that when executed by one or more computing devices, cause the one or more computing devices to:

receive a plurality of packets;

determine that at least a portion of the plurality of packets have packet header field values corresponding to a packet filtering rule;

responsive to determining that the at least a portion of the plurality of packets have packet header field values corresponding to the packet filtering rule, apply an operator specified by the packet filtering rule to the at least a portion of the plurality of packets, wherein the operator specifies one or more application header field value criteria;

determine that one or more of the at least a portion of the plurality of packets have one or more application header field values corresponding to the one or more application header field value criteria specified by the operator; and

responsive to determining that the one or more of the at least a portion of the plurality of packets have one or more application header field values corresponding to the one or more application header field criteria specified by the operator, apply at least one packet transformation function specified by the operator to the one or more of the at least a portion of the plurality of packets.

* * * * *