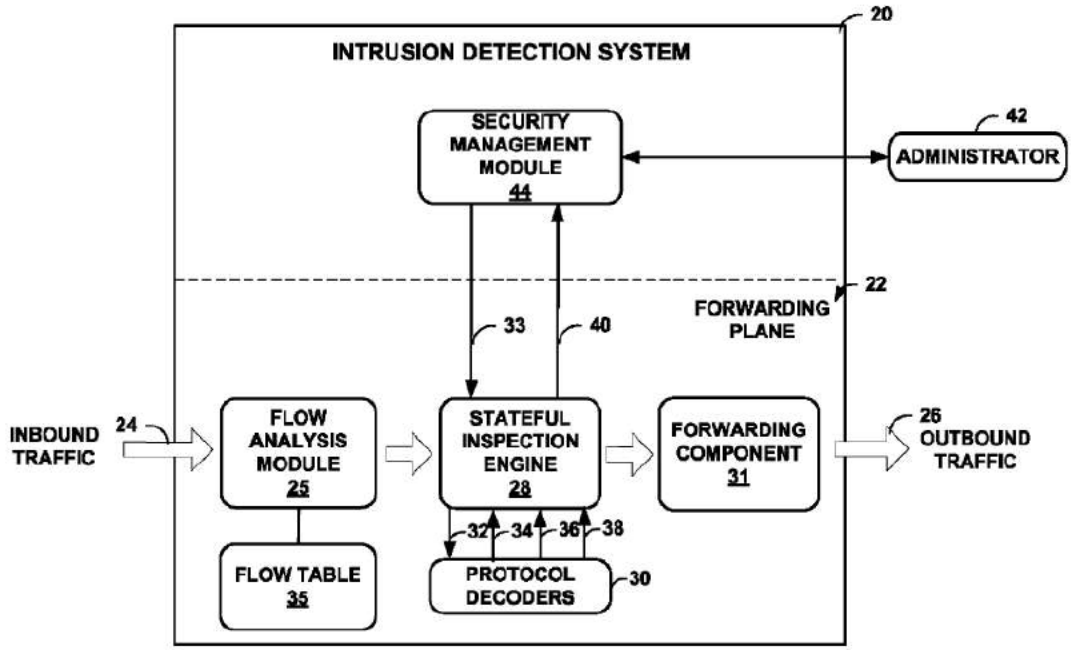
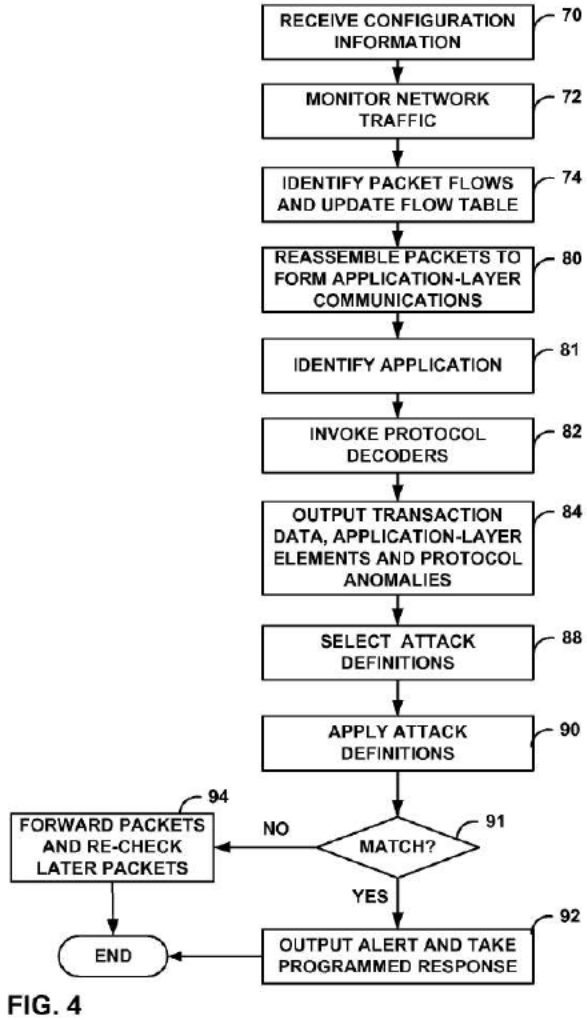
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions. Moreover, security management module **44** may present a user interface by which administrator **42** may modify assumptions regarding packet flow characteristics, such as the highest priority packet flows for monitoring, port bindings for applications, or other features of determining a type of application and protocol associated with the packet flow. Flow analysis module **25** receives inbound traffic **24** and identifies network flows within the traffic. Each network flow represents a flow of packets in one direction within the network traffic and is identified by at least a source address, a destination address and a communication protocol. Flow analysis module **25** may utilize additional information to specify network flows, including source media access control ("MAC") address, destination MAC address, source port, and destination port. Other embodiments may use other information to identify network flows, such as IP addresses. Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Yang, 7:16-60.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
1[a]. receiving, by the user's device, a request for retrieval of at least one data packet from the user's device, wherein the user's device is configured to provide a response corresponding to the received request;	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discusses analyzing "HTTP traffic" and "HTTP form data," which are associated with web pages. The webpages may an input request for a user.  **FIG. 2** (Yang, Fig. 2.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 FIG. 4 (Yang, Fig. 4.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"FIG. 6 is an example user interface **100** presented by an IDS with which an administrator interacts to create a pattern for detecting attacks. In this example, user interface **100** allows the administrator to define a compound network attack definition. As illustrated in FIG. 6, user interface **100** includes an input **102** by which the administrator may provide a name for the compound network attack. In addition, interface **100** includes an input **104** by which the administrator may provide a brief description for the compound network attack. The example of FIG. 6 illustrates a compound network attack for detecting directory traversal attempts against an Apache web server. In addition, interface **100** includes inputs **106** by which the administrator may provide other data for the compound network attack, such as a severity level **108**, keywords for use in indexing and analysis, and other data. FIG. 7 is an example of user interface **110** presented when the user selects the "Signatures" tab of user interface **100** (FIG. 6). As illustrated, user interface **110** includes an input **112** with which the administrator specifies a protocol to which the compound attack definition pertains. In this example, the administrator has selected HTTP as this exemplary compound network attack definition relates to an Apache web server. User interface **110** further includes an input **111** by which the administrator selects from a list of contexts for the particular protocol selected in input **112**. For example, input **111** provides a drop-down menu by which the administrator selects HTTP contexts, such as HTTP Authorization, HTTP data, HTTP form data and other contexts available from the headers in an HTTP communication. User interface **110** further includes an input **114** by which the administrator selects a scope for the compound attack definition is to be applied. In particular, input **114** allows the administrator to specify whether the attack is session based (i.e., to be applied over the life-time of a session and across all transactions associated with the session) or transaction based (i.e., applied on a per transaction basis and cleared at the end of each transaction).

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	User interface **110** further includes an input area **118** by which the administrator defines the constituent members of the compound attack definition. In particular, the administrator may add patterns by selecting Add button **120** or protocol anomalies by selecting Add button **122**. Input area **118** lists the members currently defined for the compound attack definition. In this example, two patterns **124**, **126** have been defined in the form of regular expressions. User interface **110** also includes an input **130** by which the administrator controls whether the constituent members must be detected in the order presented within input area **118** or whether the members may be detected in any order within the scope defined by input area **114**." (Yang, 15:11-62.)
1[b]. determining, by the remote server, at least one communication data type of the at least one data packet corresponding to the received request, wherein the at least one communication data type is determined in accordance with characteristics of the communication data packet, and wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during communication between the remote server and the user's device;	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses identifying an application or protocol based on a TCP port in a packet header without decrypting the packet. The packet may be associated with an input request for a user. Yang's IDS monitors traffic in "real-time." "In general, IDS **10** identifies packet flows in the monitored traffic, and transparently reassembles application-layer communications from the packet flows. A set of protocol-specific decoders within the IDS **10** analyzes the application-layer communications and identifies application-layer transactions. In general, a "transaction" refers to a bounded series of related application-layer communications between peer devices. For example, a single TCP connection can be used to send (receive) multiple HyperText Transfer Protocol (HTTP) requests (responses). As one example, a single web-page comprising multiple images and links to HTML pages may be fetched using a single TCP connection. An HTTP decoder identifies each request/response within the TCP connection as a different transaction. This may be useful to prevent certain attack definitions from being applied across transaction boundaries. In one embodiment, a transaction may be identified according to source and destination IP address, protocol, and source and destination port

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	numbers. Other embodiments may identify a transaction in other ways, for example, by using media access control ("MAC") addresses." (Yang, 5:1-20.) "In addition, IDS 10 monitors for traffic flowing in the opposite direction to intercept any response from the server. That is, IDS 10 identifies any response packet flow from the server, which typically utilizes the same or substantially similar network addresses, ports and/or protocol as the initial client request. Upon intercepting the response packet flow, IDS 10 performs an additional application identification test to confirm whether its initial assessment of the type of software application and protocol for the particular client-server communication session was correct. If not, IDS 10 may reclassify the communication as a different type of software application and protocol, and select and apply additional attack definitions based on the updated classification. That is, IDS 10 may retrieve the cached assessment of the application and then use the response packet flow to either confirm or modify the assessment, again caching the result." (Yang, 6:22-37.) "These techniques may allow IDS **10** to provide improved attack detection and prevention in environments where some applications no longer use a static port mapping but instead use dynamic or randomized port assignments in an attempt to evade detection and containment. For example, instant messaging programs may initiate communication on one port, but then agree to communicate thereafter on a distinct port. Other programs, such as peer-to-peer ("P2P") programs and hacker toolkits such as Metasploit, may determine ports dynamically as well. As described, IDS **10** may make an initial determination for the application type of a packet flow from a client device of public network **6** to a server device, e.g. node **8**A, based upon characteristics of the communication in addition to specified port number. IDS **10** may then analyze the return communication from node **8**A to the client device to verify the initial assumption or to modify the initial determination if necessary. In some embodiments, enterprise network **5** may include multiple IDSs **10** and **14** located within different regions (e.g., sub-networks) of enterprise network **5**. Security management

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	device **18** may operate as a central device for managing IDSs **10** and **14**. Although the example illustrated in FIG. 1 is described in terms of dedicated IDSs **10** and **14**, the functionality described herein may be incorporated within other devices, such as firewall **9** or switch **19**." (Yang, 6:45-7:2.) "FIG. 2 is a block diagram illustrating an example embodiment of an IDS **20**. In the illustrated example, IDS **20** includes a forwarding plane **22** that transparently monitors inbound network traffic **24** and forwards the network traffic as outbound network traffic **26**. In the example illustrated by FIG. 2, forwarding plane **22** includes flow analysis module **25**, stateful inspection engine **28**, protocol decoders **30**, forwarding component **31** and security management module **44**. Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions. Moreover, security management module **44** may present a user interface by which administrator **42** may modify assumptions regarding packet flow characteristics, such as the highest priority packet flows for monitoring, port bindings for applications, or other features of determining a type of application and protocol associated with the packet flow. Flow analysis module **25** receives inbound traffic **24** and identifies network flows within the traffic. Each network flow represents a flow of packets in one direction within the network traffic and is identified by at least a source address, a destination address and a communication protocol. Flow analysis module **25** may utilize additional information to specify network flows, including source media access control ("MAC") address,

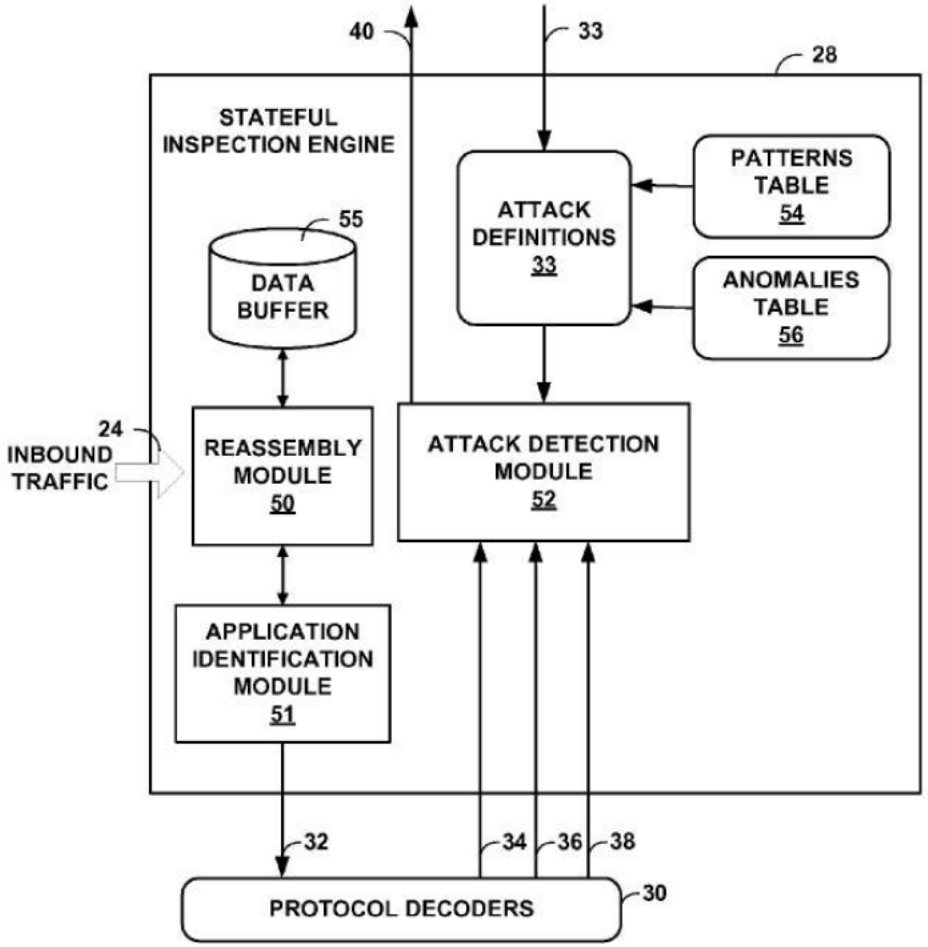
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	destination MAC address, source port, and destination port. Other embodiments may use other information to identify network flows, such as IP addresses. Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Yang, 7:16-60.) "IDS **20** may use a minimum data size of the reassembled TCP segments, in addition to the signature, in order to identify the types of applications. Certain applications may require a minimum amount of data, so IDS **20** may distinguish malicious packet flows by determining whether the packet flow contains enough data for the identified protocol. Moreover, IDS **20** may not necessarily recognize every application. In one embodiment, when an application is unknown, IDS **20** may simply forward the packet flow. If IDS **20** cannot identify a given application, it may be because that application is not a typical target for a malicious packet flow. Other embodiments may take other actions for unidentified applications, however, such as discarding all packets which target unknown applications or applying a default signature to all packet flows associated with unknown application types. Other embodiments may also utilize other protocols, such as the user datagram protocol (UDP); IDS **20** accordingly may require a minimum data size of UDP segments in order to identify the application associated with the UDP segments. For each packet flow, stateful inspection engine **28** buffers a copy of the packet flow and reassembles the buffered packet flow to form application-layer communications **32**. For example, stateful inspection engine **28** may reconstruct TCP segments into application-layer communications **32**, which represent protocol-specific messages.

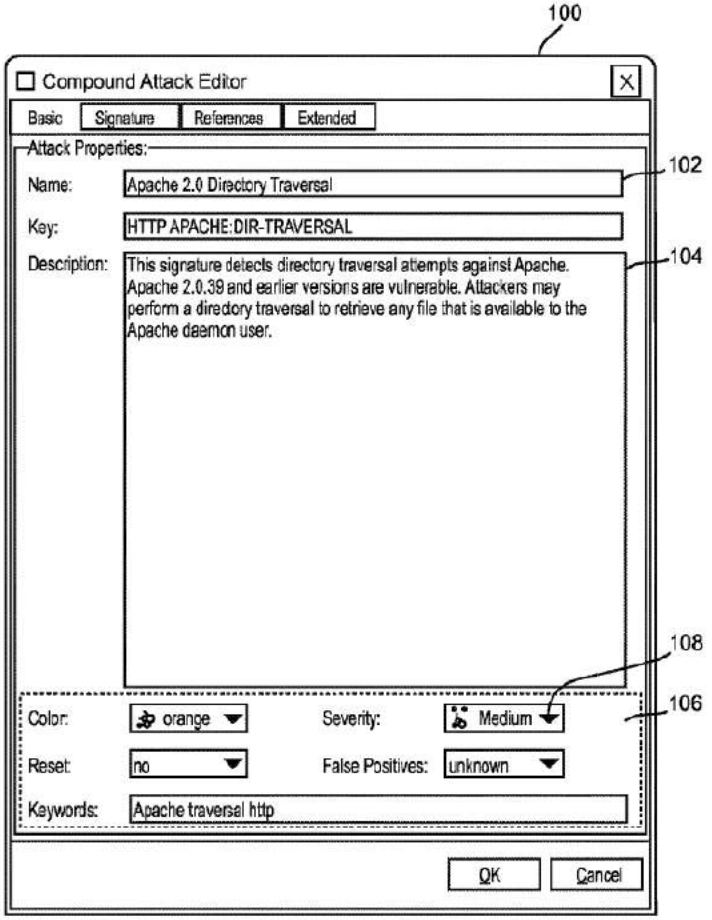
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	Stateful inspection engine **28** invokes the appropriate one of protocol decoders **30** based on the identified type of application determination to analyze the application-layer communications **32**. Protocol decoders **30** represent a set of one or more protocol-specific software modules. Each of protocol decoders **30** corresponds to a different communication protocol or service. Examples of communication protocols that may be supported by protocol decoders **30** include the HyperText Transfer Protocol ("HTTP"), the File Transfer Protocol ("FTP"), the Network News Transfer Protocol ("NNTP"), the Simple Mail Transfer Protocol ("SMTP"), Telnet, Domain Name System ("DNS"), Gopher, Finger, the Post Office Protocol ("POP"), the Secure Socket Layer ("SSL") protocol, the Lightweight Directory Access Protocol ("LDAP"), Secure Shell ("SSH"), Server Message Block ("SMB") and other protocols." (Yang, 8:8-48.) "Application identification module **51** identifies the type of application and underlying protocol for each intercepted communication session. When stateful inspection engine **28** receives a packet as part of a packet flow, reassembly module **50** buffers the packet in data buffer **55**. In one embodiment, data buffer **55** may store data as a sliding window. That is, data buffer **55** may store data until becoming full or reaching a specified required amount of minimum data for identification. When full, data buffer **55** discards certain data to make room for storing new data. In one embodiment, data buffer **55** may store and discard data according to a first-in, first-out ("FIFO")-like protocol wherein the first data to be stored is the first data to be discarded when data buffer **55** becomes full. In another embodiment, data buffer **55** may discard data according to a least recently used protocol wherein, when data buffer **55** is full, the packet flow which has been least recently used will be discarded to make room for new data to be stored. In one embodiment, reassembly module **50** may associate packets in a packet flow, and packet flows as a communication session, according to the 5-tuple {source IP address, destination IP address, protocol, source port, destination port}. Other embodiments may use other forms of associating packets. For example, in one embodiment, IDS **20** may be part of a network that utilizes virtual local area networks (VLANs).

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	Accordingly, reassembly module **50** may associate packets in a packet flow according to a VLAN identifier, a source address, and a destination address. In any case, reassembly module **50** may utilize the information maintained within flow table **35** (FIG. 2) to reassemble network data, e.g., to form reassembled TCP data. Application identification module **51** analyzes the reassembled data for the packet flow to identify the type of application and protocol associated with the packet flow. If application identification module **51** is unable to identify the type of application and protocol associated with the packet flow, application identification module **51** may use the well-known static port binding as a default application selection. Table 1, below, shows an example static port binding list. Other embodiments may use more, fewer, or different entries in a static port table. Moreover, administrator **42** may configure the static port mapping using security management module **44**." (Yang, 9:27-10:3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	TABLE I PORT / APPLICATION 20 FTP 22 SSH 23 Telnet 25 SMTP 43 WHOIS 53 DNS 67 BOOTP or DHCP 70 Gopher 79 Finger 80 HTTP 109 POP 110 POP3 113 ident/IRC 118 SQL 119 NNTP 194 IRC 443 HTTPS 445 SMB 564 RTSP (Yang, Table 1.) "In the case where application identification module **51** is unable to identify a type of application and protocol for a packet flow, application identification module **51** may use the default static port mapping to determine an application, causing IDS **20** to respond accordingly. In some cases, application identification module **51** may not be able to identify the application and the static port mapping may not have an entry for the requested port number. Various embodiments may treat this situation according to specifications of, for example, a system administrator. For example, in one embodiment, IDS **20** simply

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	forwards packet flows with undetermined application types and protocols that cannot be determined by the static port mapping pass, as an unknown application may indicate that the packet flow is not targeting any type of application known to pose a security threat. In other embodiments, IDS **20** may automatically discard packet flows with unknown application types and protocols that cannot be determined by the static port mapping." (Yang, 10:25-42.) "Once configured, IDS **20** monitors network traffic **24** (**72**). In some configurations, stateful inspection engine **28** of forwarding plane **22** may receive network traffic and mirror the network traffic for purposes of analysis. Forwarding component **31** seamlessly forwards the original network traffic. In other embodiments, traffic is not mirrored, rather a line-rate buffering approach is used to analyze the traffic in real-time prior to forwarding." (Yang, 11:63-12:3.) "4. The method of claim 2, wherein the compared properties comprise at least one of a packet flow pattern, a destination port number, a source port number, and a protocol." (Yang, 17:3-5.) *See* also claim element 17[e].
1[c]. receiving, by the user's device, a privacy preference for the user's device, wherein the privacy preference comprises a list of allowed data packet communication types for sharing during communication;	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses attack definitions to detect and prevent network attacks and other security risks. If an attack is not detected, Yang's IDS would allow the data packets to pass. These definitions may be edited by an administrator and provided to a user's device.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 **FIG. 3** (Yang, Fig. 3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 **FIG. 6** (Yang, Fig. 6.)

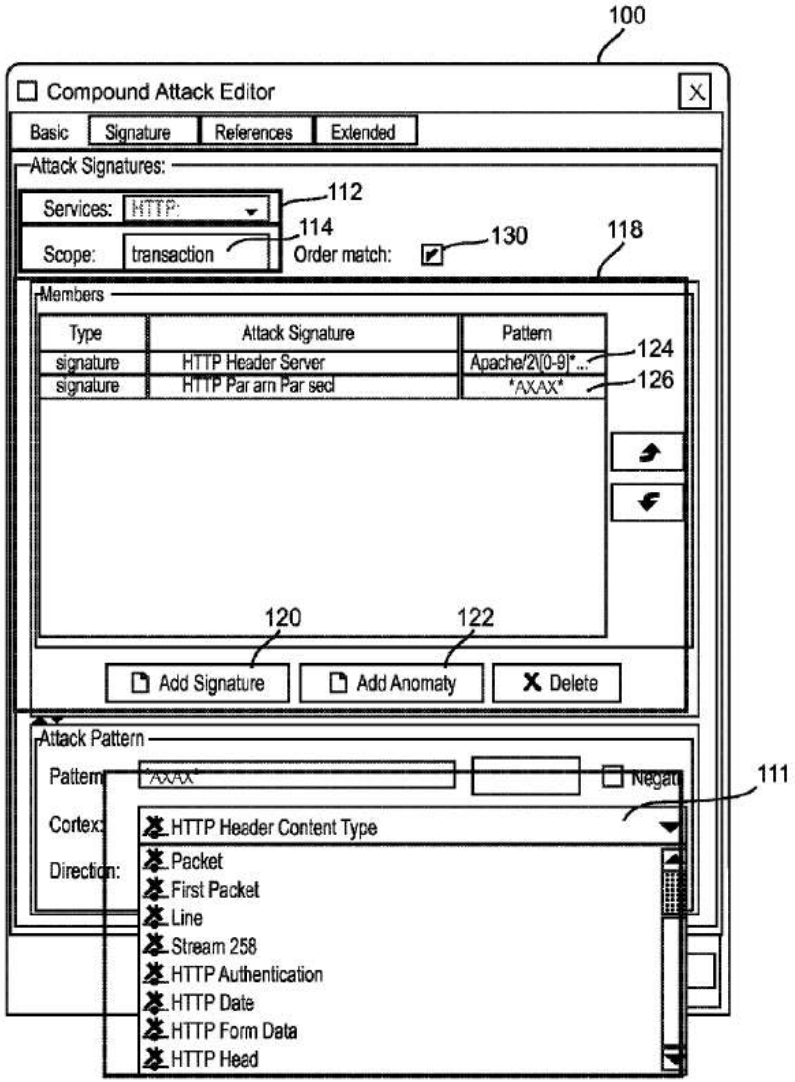


FIG. 7

(Yang, Fig. 7.)

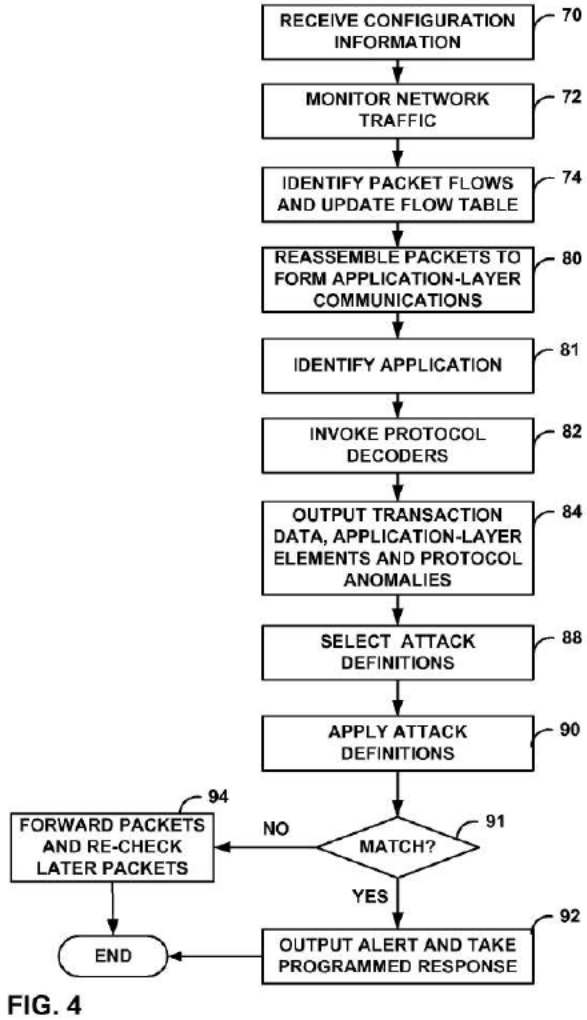
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"In accordance with the principles of the invention, IDS **10** integrates pattern matching with application- and protocol-specific anomaly detection to identify sophisticated attack behaviors. In one embodiment, IDS **10** allows the system administrator to specify attack definitions. In one embodiment, the system administrator may specify compound attack definitions. Further details on application of attack definitions, e.g. compound attack definitions, may be found within U.S. patent application Ser. No. 11/045,572, Guruswamy et al., "Compound Attack Detection in a Computer Network," filed Jan. 27, 2005, assigned to the assignee of the current application, which is incorporated herein by reference in its entirety. In general, the attack definitions may specify, for example, any combination of textual and non-textual (e.g., binary) patterns and protocol anomalies to define complex attack signatures. Moreover, IDS **10** may associate particular signatures with protocols of certain applications. For a given communication session intercepted by IDS **10**, the IDS attempts to identify the application type and underlying protocol for the packet flows of the session in order to select one or more attack signatures to apply to the packet flows." (Yang, 4:52-67.) "IDS **10** applies the attack definitions to the elements and the protocol-specific anomalies identified by the protocol decoders to detect and prevent network attacks. For example, a system administrator may specify a compound network attack that includes the protocol anomaly of repeated FTP login failure and a pattern that matches a login username of "root." In this manner, the system administrator may combine pattern analysis with protocol anomalies to define complex attack definitions. In the event of a network attack, IDS **10** may take one or more programmed actions, such as automatically dropping packet flows associated with the application-layer communications within which the network attack was detected." (Yang, 5:53-65.) "For example, upon first receiving a connection request from a client and destined for a server, IDS **10** may use a static port number list to make an initial assessment of the type

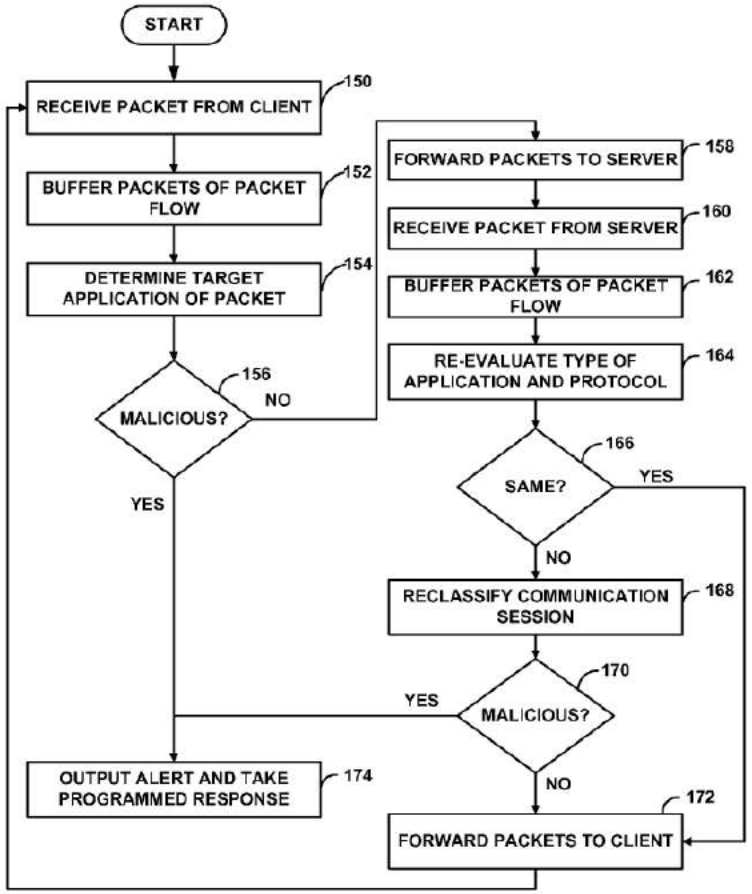
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	of software application and protocol associated with the received packet flow. As one example, HTTP packets are typically directed to port **80**, while e-mail packets have been directed to port **25**. Therefore, IDS **10** may utilize the static port assignments to make an initial assessment for the type of software application and protocol associated with the received packet flow. Based on this initial assessment, IDS selects and applies a first set of patterns, i.e., attack definitions. This initial set may include protocol-specific attack definitions as well as default attack definitions. In addition, IDS **10** monitors for traffic flowing in the opposite direction to intercept any response from the server. That is, IDS **10** identifies any response packet flow from the server, which typically utilizes the same or substantially similar network addresses, ports and/or protocol as the initial client request. Upon intercepting the response packet flow, IDS **10** performs an additional application identification test to confirm whether its initial assessment of the type of software application and protocol for the particular client-server communication session was correct. If not, IDS **10** may reclassify the communication as a different type of software application and protocol, and select and apply additional attack definitions based on the updated classification. That is, IDS **10** may retrieve the cached assessment of the application and then use the response packet flow to either confirm or modify the assessment, again caching the result. Moreover, IDS **10** may apply the newly selected attack definitions to the first packet flow, e.g. the packet flow initially received from the client. That is, IDS **10** may include buffering capability to temporarily store the packet flow on which the initial assessment of application type was based, and apply any newly selected attack definitions to the buffered packet flow." (Yang, 6:9-44.) "Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions. Moreover, security management module **44** may present a user interface by which administrator **42** may modify assumptions regarding packet flow characteristics, such as the highest priority packet flows for monitoring, port bindings for applications, or other features of determining a type of application and protocol associated with the packet flow." (Yang, 7:24-38.) "Stateful inspection engine **28** receives transaction data **34**, application-layer elements **36** and protocol anomaly data **38** from protocol decoders **30**. Stateful inspection engine **28** applies attack definitions **33** to protocol-specific application-layer elements **36** and anomaly data **38** to detect and prevent network attacks and other security risks." (Yang, 8:55-60.) "FIG. 3 is a block diagram that illustrates an example embodiment of stateful inspection engine **28**. In the example embodiment, stateful inspection engine **28** includes reassembly module **50**, application identification module **51**, and attack detection module **52**. In addition, stateful inspection engine **28** includes patterns table **54**, data buffer **55**, anomalies table **56**, and attack definitions **33**. Reassembly module **50** receives inbound network traffic **24** and reassembles application-layer communications **32** from the packet flows. Reassembly module **50** forwards the reassembled application-layer communications **32** to the appropriate protocol decoders **30** for processing. Stateful inspection engine **28** stores attack definitions **33** received from security management module **44**. Attack definitions **33** may be stored, for example, in a computer-readable medium, such as random access memory ("RAM"). Each of attack definitions **33** specifies a combination of one or more patterns specified within patterns

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	table **54** and one or more protocol-specific anomalies specified within anomalies table **56**." (Yang, 9:6-26.) "Initially, security management module **44** receives configuration information from administrator **42** and, in response, configures IDS **20** to monitor a network or portions thereof (subnets) of interest (**70**). During this process, configuration manager **44** may present a user interface by which administrator **42** specifies patterns or other attack definitions **33**." (Yang, 11:56-62.) "FIG. 6 is an example user interface **100** presented by an IDS with which an administrator interacts to create a pattern for detecting attacks. In this example, user interface **100** allows the administrator to define a compound network attack definition. As illustrated in FIG. 6, user interface **100** includes an input **102** by which the administrator may provide a name for the compound network attack. In addition, interface **100** includes an input **104** by which the administrator may provide a brief description for the compound network attack. The example of FIG. 6 illustrates a compound network attack for detecting directory traversal attempts against an Apache web server. In addition, interface **100** includes inputs **106** by which the administrator may provide other data for the compound network attack, such as a severity level **108**, keywords for use in indexing and analysis, and other data. FIG. 7 is an example of user interface **110** presented when the user selects the "Signatures" tab of user interface **100** (FIG. 6). As illustrated, user interface **110** includes an input **112** with which the administrator specifies a protocol to which the compound attack definition pertains. In this example, the administrator has selected HTTP as this exemplary compound network attack definition relates to an Apache web server. User interface **110** further includes an input **111** by which the administrator selects from a list of contexts for the particular protocol selected in input **112**. For example, input **111** provides a drop-down menu by which the administrator selects HTTP

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	contexts, such as HTTP Authorization, HTTP data, HTTP form data and other contexts available from the headers in an HTTP communication. User interface **110** further includes an input **114** by which the administrator selects a scope for the compound attack definition is to be applied. In particular, input **114** allows the administrator to specify whether the attack is session based (i.e., to be applied over the life-time of a session and across all transactions associated with the session) or transaction based (i.e., applied on a per transaction basis and cleared at the end of each transaction). User interface **110** further includes an input area **118** by which the administrator defines the constituent members of the compound attack definition. In particular, the administrator may add patterns by selecting Add button **120** or protocol anomalies by selecting Add button **122**. Input area **118** lists the members currently defined for the compound attack definition. In this example, two patterns **124**, **126** have been defined in the form of regular expressions. User interface **110** also includes an input **130** by which the administrator controls whether the constituent members must be detected in the order presented within input area **118** or whether the members may be detected in any order within the scope defined by input area **114**." (Yang, 15:11-62.) *See* also claim elements 1[b] and 1[d].
1[d]. modifying data packets corresponding to requests for sharing of responses that are not compatible with the received privacy preference; and	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses blocking communications (*e.g.*, dropping packets) in accordance with a determination of a security risk.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 FIG. 4 (Yang, Fig. 4.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 FIG. 5 (Yang, Fig. 5.)

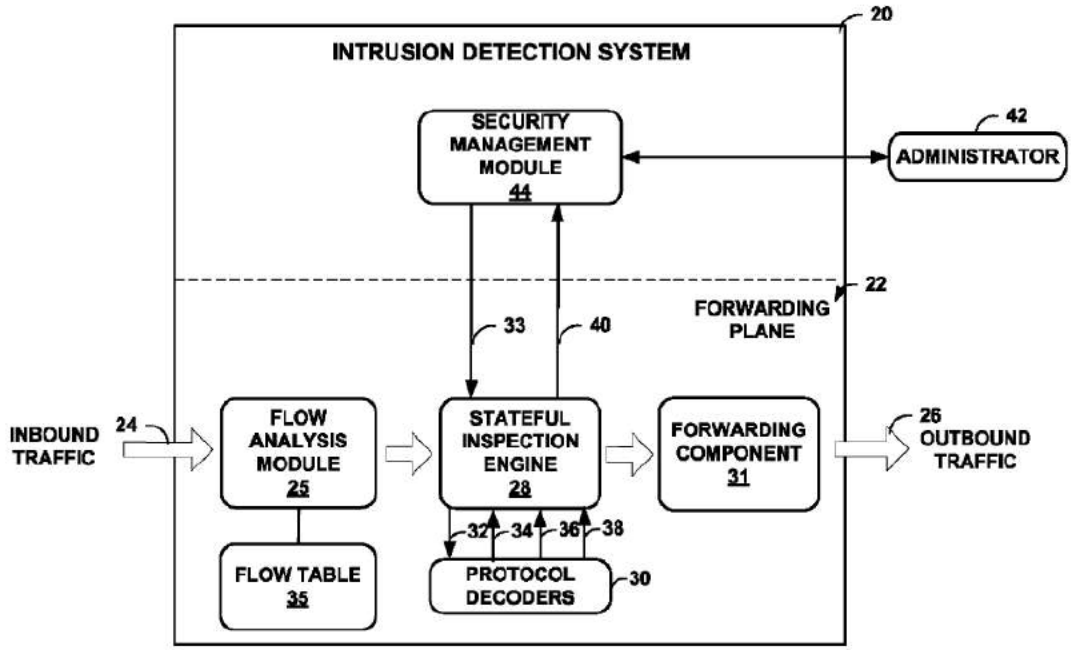
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"IDS **10** applies the attack definitions to the elements and the protocol-specific anomalies identified by the protocol decoders to detect and prevent network attacks. For example, a system administrator may specify a compound network attack that includes the protocol anomaly of repeated FTP login failure and a pattern that matches a login username of "root." In this manner, the system administrator may combine pattern analysis with protocol anomalies to define complex attack definitions. In the event of a network attack, IDS **10** may take one or more programmed actions, such as automatically dropping packet flows associated with the application-layer communications within which the network attack was detected. In accordance with the techniques described herein, IDS **10** may attempt to identify the software application and protocol associated with each communication session, i.e., each corresponding pair of packet flows between a given client and server. IDS **10** uses the identified type of software application and underlying protocol type for the communication session to control selection of any protocol-specific patterns to be applied to the packet flows. IDS **10** may temporarily cache the initial assessment of the software application for later comparison and confirmation." (Yang, 5:53-6:8.) "Flow analysis module **25** receives inbound traffic **24** and identifies network flows within the traffic. Each network flow represents a flow of packets in one direction within the network traffic and is identified by at least a source address, a destination address and a communication protocol. Flow analysis module **25** may utilize additional information to specify network flows, including source media access control ("MAC") address, destination MAC address, source port, and destination port. Other embodiments may use other information to identify network flows, such as IP addresses. Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow

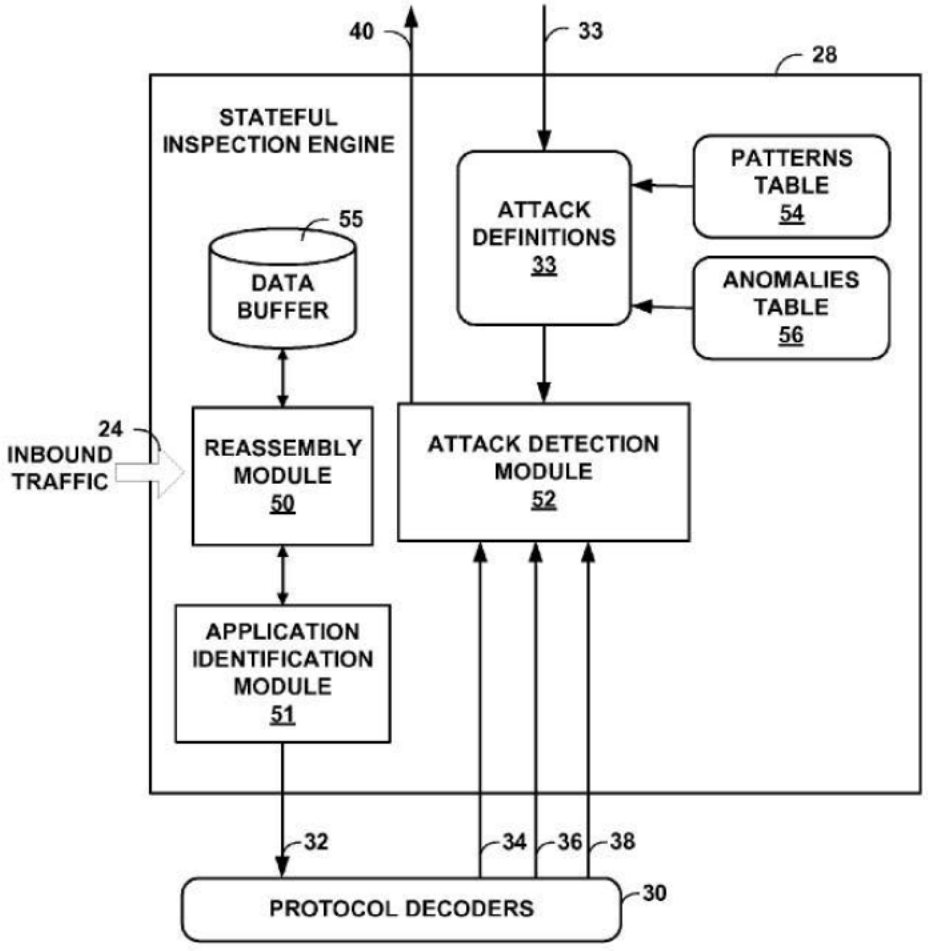
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Yang, 7:39-60.) "To detect an attack or other malicious activity, attack detection module **52** applies attack definitions **33** to application-layer elements **36** and protocol anomaly data **38** received from protocol decoders **30**. In particular, for each of attack definitions **33**, attack detection module **52** selects the one or more patterns within patterns table **52** specified by the attack definition and determines whether any of application-layer elements **36** match the defined patterns. Each of the patterns may be defined as a respective "regular expression," which generally refers to a formula that is used to match patterns within data. In addition to determining whether the defined patterns are present, attack detection module **52** may determine whether any protocol anomalies detected by protocol decoders **30** match the protocol anomalies specified by attack definitions **33**. Attack detection module **52** determines that the corresponding packet flow matches one of attack definitions **33** when both the patterns and protocol anomalies specified by the compound attack definition are detected within a given communication session. Further, each of attack definitions **33** may specify whether the pattern matching and protocol anomalies must be satisfied on a per-transaction basis or over the lifetime of the communication session." (Yang, 11:20-30.) "Upon receiving data from protocol decoders **30**, stateful inspection engine **28** selects the attack definitions **33** that are defined for the corresponding protocol, and optionally a set of default attack definitions when no particular type of application has been identified (**88**). Stateful inspection engine **28** then applies the selected compound attack definitions to determine whether the communication session represents a security risk (**90**). When applying a given compound attack definition, stateful inspection engine **28** determines whether all of the specified patterns and protocol anomalies are satisfied for any given communication session between peers, either on a per-transaction basis or over the lifetime

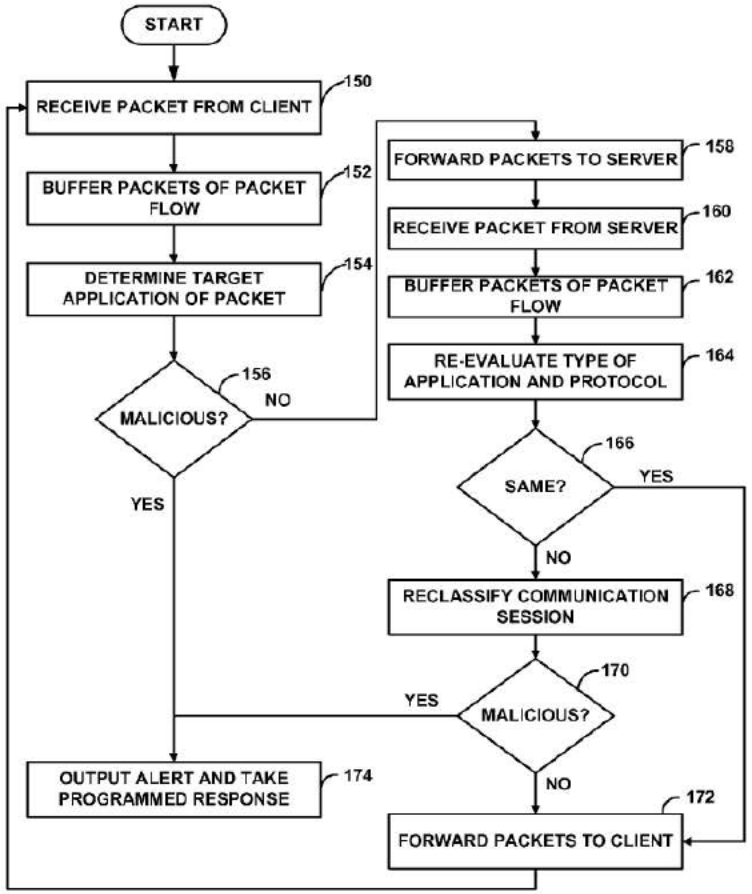
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	of the communication session, as specified by the compound attack definition. Moreover, if required by the compound attack definition, stateful inspection engine **28** may determine whether the specified patterns and protocol anomalies are satisfied in a required order. In the event a security risk (i.e., match) is detected (**91**), stateful inspection engine **28** outputs alert **40** to security management module **44** for logging and further analysis (**92**). In addition, stateful inspection engine **28** may take any of a number of programmed responses, such as dropping the packets associated with the communication session, automatically closing the communication session or other action. If no security risk is detected for a given application-layer communication session, forwarding component **31** forwards the packet flow to the destination (**94**). IDS **20** then waits for a response packet flow and reanalyzes the packet flow in light of the response to check the initial determination." (Yang, 12:51-13:12.) *See* also claim element 1[c].
1[e]. maintaining communication between the remote server and the user's device, with sharing of the modified data packets.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses maintaining communications between the IDS and the user's device, while blocking and allowing some communications according to whether a security risk is determined. "IDS **10** applies the attack definitions to the elements and the protocol-specific anomalies identified by the protocol decoders to detect and prevent network attacks. For example, a system administrator may specify a compound network attack that includes the protocol anomaly of repeated FTP login failure and a pattern that matches a login username of "root." In this manner, the system administrator may combine pattern analysis with protocol anomalies to define complex attack definitions. In the event of a network attack, IDS **10** may take one or more programmed actions, such as automatically dropping

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	packet flows associated with the application-layer communications within which the network attack was detected. In accordance with the techniques described herein, IDS **10** may attempt to identify the software application and protocol associated with each communication session, i.e., each corresponding pair of packet flows between a given client and server. IDS **10** uses the identified type of software application and underlying protocol type for the communication session to control selection of any protocol-specific patterns to be applied to the packet flows. IDS **10** may temporarily cache the initial assessment of the software application for later comparison and confirmation." (Yang, 5:53-6:8.) "Flow analysis module **25** receives inbound traffic **24** and identifies network flows within the traffic. Each network flow represents a flow of packets in one direction within the network traffic and is identified by at least a source address, a destination address and a communication protocol. Flow analysis module **25** may utilize additional information to specify network flows, including source media access control ("MAC") address, destination MAC address, source port, and destination port. Other embodiments may use other information to identify network flows, such as IP addresses. Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Yang, 7:39-60.) *See* also claim elements 1[pre] and 1[d].

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
2. The method of claim 1, wherein the communication data type of the at least one data packet is determined from metadata of communication with the remote server.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses identifying an application or protocol based on a TCP port in a packet header, which is metadata. *See* claim element 1[b].
3. The method of claim 1, further comprising:	*See* claim 1.
3[a]. determining, by the user's device, at least one communication data pattern of at least one data packet corresponding to the received request; and	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses protocol-specific anomaly analysis, which includes using anomaly information to detect sophisticated attack behaviors.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 **FIG. 2** (Yang, Fig. 2.)

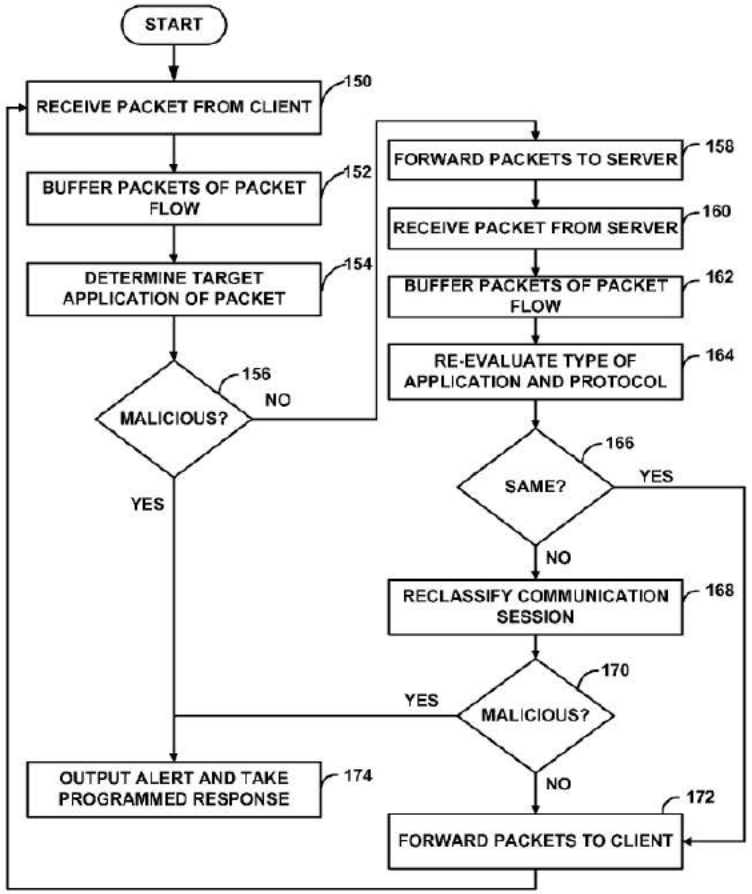
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 **FIG. 3** (Yang, Fig. 3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 FIG. 5 (Yang, Fig. 5.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"In the event a security risk is detected, stateful inspection engine **28** outputs alert **40** to security management module **44** (FIG. 2) for logging and further analysis. Stateful inspection engine **28** may also direct forwarding component **31** to automatically drop the packet flow associated with the application-layer communications within which the network attack was detected. In this manner, stateful inspection engine **28** combines pattern matching with protocol-specific anomaly analysis to detect sophisticated attack behaviors." (Yang, 11:43-51.) "Upon receiving data from protocol decoders **30**, stateful inspection engine **28** selects the attack definitions **33** that are defined for the corresponding protocol, and optionally a set of default attack definitions when no particular type of application has been identified (**88**). Stateful inspection engine **28** then applies the selected compound attack definitions to determine whether the communication session represents a security risk (**90**). When applying a given compound attack definition, stateful inspection engine **28** determines whether all of the specified patterns and protocol anomalies are satisfied for any given communication session between peers, either on a per-transaction basis or over the lifetime of the communication session, as specified by the compound attack definition. Moreover, if required by the compound attack definition, stateful inspection engine **28** may determine whether the specified patterns and protocol anomalies are satisfied in a required order." (Yang, 12:55-67.) *See* also claim elements 1[b] and 1[c].
3[b]. modifying data packets corresponding to requests for sharing of responses corresponding to at least one data pattern, that are not compatible with the received privacy preference,	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses blocking communications (*e.g.*, dropping packets) in accordance with a determination of a security risk (*e.g.*, a matching pattern). "IDS **10** applies the attack definitions to the elements and the protocol-specific anomalies identified by the protocol decoders to detect and prevent network attacks. For example, a

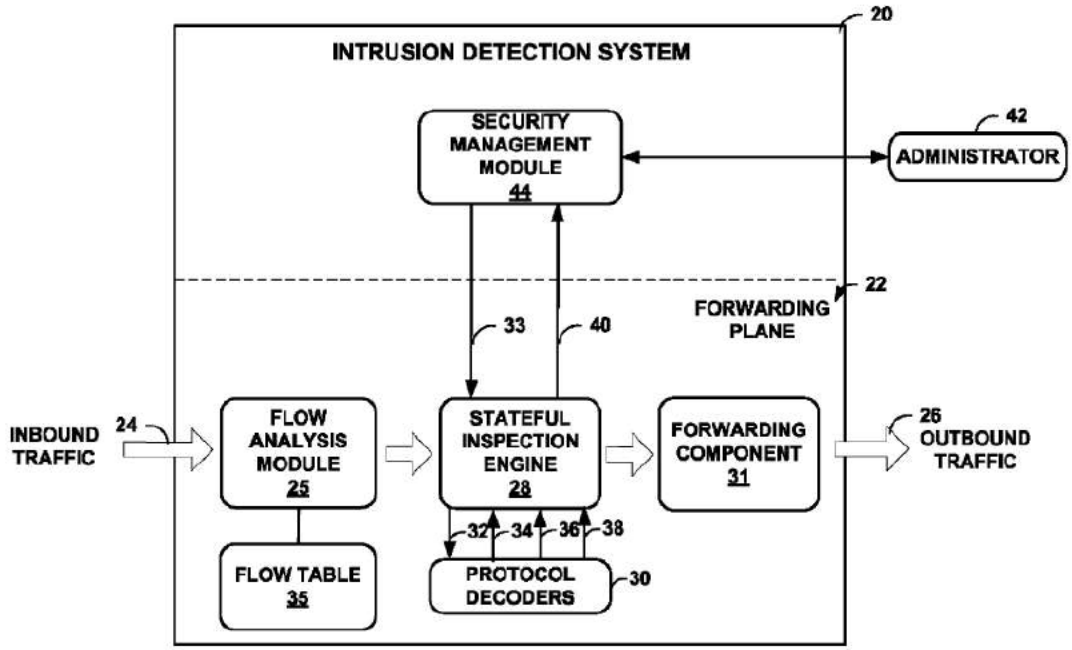
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	system administrator may specify a compound network attack that includes the protocol anomaly of repeated FTP login failure and a pattern that matches a login username of "root." In this manner, the system administrator may combine pattern analysis with protocol anomalies to define complex attack definitions. In the event of a network attack, IDS **10** may take one or more programmed actions, such as automatically dropping packet flows associated with the application-layer communications within which the network attack was detected. In accordance with the techniques described herein, IDS **10** may attempt to identify the software application and protocol associated with each communication session, i.e., each corresponding pair of packet flows between a given client and server. IDS **10** uses the identified type of software application and underlying protocol type for the communication session to control selection of any protocol-specific patterns to be applied to the packet flows. IDS **10** may temporarily cache the initial assessment of the software application for later comparison and confirmation." (Yang, 5:53-6:8.) "FIG. 2 is a block diagram illustrating an example embodiment of an IDS **20**. In the illustrated example, IDS **20** includes a forwarding plane **22** that transparently monitors inbound network traffic **24** and forwards the network traffic as outbound network traffic **26**. In the example illustrated by FIG. 2, forwarding plane **22** includes flow analysis module **25**, stateful inspection engine **28**, protocol decoders **30**, forwarding component **31** and security management module **44**. Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions. Moreover, security management

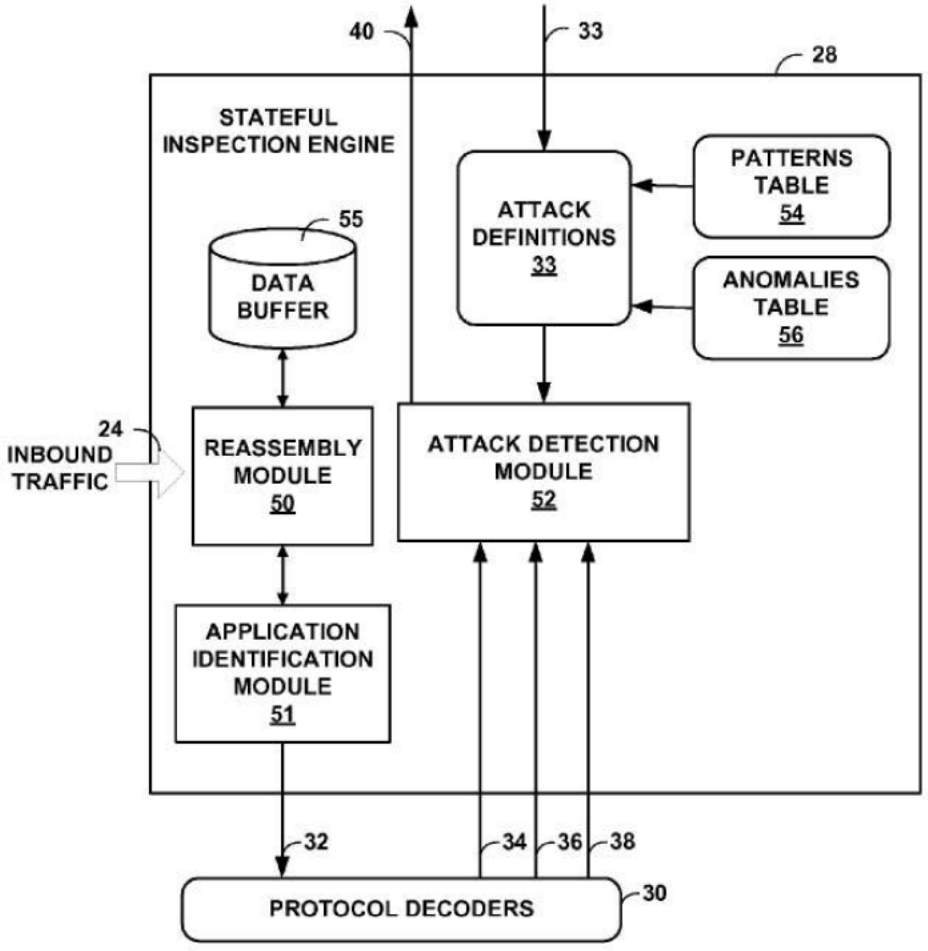
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	module **44** may present a user interface by which administrator **42** may modify assumptions regarding packet flow characteristics, such as the highest priority packet flows for monitoring, port bindings for applications, or other features of determining a type of application and protocol associated with the packet flow. Flow analysis module **25** receives inbound traffic **24** and identifies network flows within the traffic. Each network flow represents a flow of packets in one direction within the network traffic and is identified by at least a source address, a destination address and a communication protocol. Flow analysis module **25** may utilize additional information to specify network flows, including source media access control ("MAC") address, destination MAC address, source port, and destination port. Other embodiments may use other information to identify network flows, such as IP addresses. Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Yang, 7:16-60.) *See* also claim elements 1[c] and 1[d].
3[c]. wherein the privacy preference comprises a list of allowed data patterns for sharing during communication.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses blocking (*e.g.*, dropping packets) or allowing communications in accordance with a determination of a security risk (*e.g.*, a matching pattern).

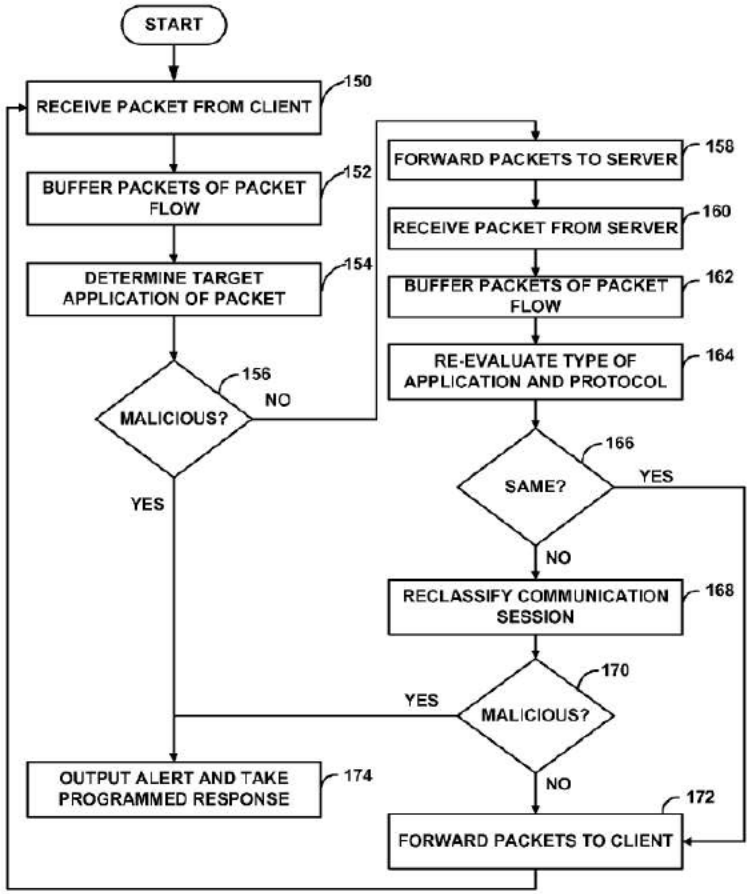
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 FIG. 5 (Yang, Fig. 5.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"Upon receiving data from protocol decoders **30**, stateful inspection engine **28** selects the attack definitions **33** that are defined for the corresponding protocol, and optionally a set of default attack definitions when no particular type of application has been identified (**88**). Stateful inspection engine **28** then applies the selected compound attack definitions to determine whether the communication session represents a security risk (**90**). When applying a given compound attack definition, stateful inspection engine **28** determines whether all of the specified patterns and protocol anomalies are satisfied for any given communication session between peers, either on a per-transaction basis or over the lifetime of the communication session, as specified by the compound attack definition. Moreover, if required by the compound attack definition, stateful inspection engine **28** may determine whether the specified patterns and protocol anomalies are satisfied in a required order. In the event a security risk (i.e., match) is detected (**91**), stateful inspection engine **28** outputs alert **40** to security management module **44** for logging and further analysis (**92**). In addition, stateful inspection engine **28** may take any of a number of programmed responses, such as dropping the packets associated with the communication session, automatically closing the communication session or other action. If no security risk is detected for a given application-layer communication session, forwarding component **31** forwards the packet flow to the destination (**94**). IDS **20** then waits for a response packet flow and reanalyzes the packet flow in light of the response to check the initial determination." (Yang, 12:51-13:12.) *See* also claim elements 1[c], 3[a], and 3[b].
4. The method of claim 3, wherein the communication data pattern is determined based on a range selected from a group consisting of: data packet frequency, data packet size, data packet speed, data packet count, data packet ratio compared to other	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses protocol-specific anomaly analysis, which includes using anomaly information to detect sophisticated attack behaviors. Attack definition for the analysis indicates patterns, which may be a required order or repetitions.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
data type flow, data packet repetition, and data packet order.	*See* claim elements 1[b], 1[c], 1[d], and claim 3.
5. The method of claim 1, wherein the communication data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, encoding method, keywords, driver, and communication protocol.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses identifying an application or protocol based on a TCP port in a packet header. *See* claim element 1[b].
6. The method of claim 1, further comprising:	*See* claim 1.
6[a]. determining, by the remote server, a communication data pattern of the at least one data packet of the received request; and	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses protocol-specific anomaly analysis, which includes using anomaly information to detect sophisticated attack behaviors.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 **FIG. 2** (Yang, Fig. 2.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 **FIG. 3** (Yang, Fig. 3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	 FIG. 5 (Yang, Fig. 5.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"In the event a security risk is detected, stateful inspection engine **28** outputs alert **40** to security management module **44** (FIG. 2) for logging and further analysis. Stateful inspection engine **28** may also direct forwarding component **31** to automatically drop the packet flow associated with the application-layer communications within which the network attack was detected. In this manner, stateful inspection engine **28** combines pattern matching with protocol-specific anomaly analysis to detect sophisticated attack behaviors." (Yang, 11:43-51.) "Upon receiving data from protocol decoders **30**, stateful inspection engine **28** selects the attack definitions **33** that are defined for the corresponding protocol, and optionally a set of default attack definitions when no particular type of application has been identified (**88**). Stateful inspection engine **28** then applies the selected compound attack definitions to determine whether the communication session represents a security risk (**90**). When applying a given compound attack definition, stateful inspection engine **28** determines whether all of the specified patterns and protocol anomalies are satisfied for any given communication session between peers, either on a per-transaction basis or over the lifetime of the communication session, as specified by the compound attack definition. Moreover, if required by the compound attack definition, stateful inspection engine **28** may determine whether the specified patterns and protocol anomalies are satisfied in a required order." (Yang, 12:55-67.) *See* also claim elements 1[b] and 1[c].
6[b]. linking at least one response from the user's device to a type of data packet from the user's device, wherein the linking is based on the communication data type and communication data pattern of the at least one response.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses that in processing the reply packet, the IDS's analysis would receive the currently identified protocol type (*e.g.*, HTTP) and make a new protocol type identification (*e.g.*, FTP). Yang further discloses applying an "attack definition" to identify a packet that is "malicious" (or not). Yang also describes how the initial identification of an application or protocol is, itself, based in part on pattern information.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"IDS **10** applies the attack definitions to the elements and the protocol-specific anomalies identified by the protocol decoders to detect and prevent network attacks. For example, a system administrator may specify a compound network attack that includes the protocol anomaly of repeated FTP login failure and a pattern that matches a login username of "root." In this manner, the system administrator may combine pattern analysis with protocol anomalies to define complex attack definitions. In the event of a network attack, IDS **10** may take one or more programmed actions, such as automatically dropping packet flows associated with the application-layer communications within which the network attack was detected. In accordance with the techniques described herein, IDS **10** may attempt to identify the software application and protocol associated with each communication session, i.e., each corresponding pair of packet flows between a given client and server. IDS **10** uses the identified type of software application and underlying protocol type for the communication session to control selection of any protocol-specific patterns to be applied to the packet flows. IDS **10** may temporarily cache the initial assessment of the software application for later comparison and confirmation." (Yang, 5:53-6:8.) "IDS **20** may use a minimum data size of the reassembled TCP segments, in addition to the signature, in order to identify the types of applications. Certain applications may require a minimum amount of data, so IDS **20** may distinguish malicious packet flows by determining whether the packet flow contains enough data for the identified protocol. Moreover, IDS **20** may not necessarily recognize every application. In one embodiment, when an application is unknown, IDS **20** may simply forward the packet flow. If IDS **20** cannot identify a given application, it may be because that application is not a typical target for a malicious packet flow. Other embodiments may take other actions for unidentified applications, however, such as discarding all packets which target unknown applications or applying a default signature to all packet flows associated with unknown application types. Other embodiments may also utilize other protocols, such as the user

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	datagram protocol (UDP); IDS **20** accordingly may require a minimum data size of UDP segments in order to identify the application associated with the UDP segments. For each packet flow, stateful inspection engine **28** buffers a copy of the packet flow and reassembles the buffered packet flow to form application-layer communications **32**. For example, stateful inspection engine **28** may reconstruct TCP segments into application-layer communications **32**, which represent protocol-specific messages. Stateful inspection engine **28** invokes the appropriate one of protocol decoders **30** based on the identified type of application determination to analyze the application-layer communications **32**. Protocol decoders **30** represent a set of one or more protocol-specific software modules. Each of protocol decoders **30** corresponds to a different communication protocol or service. Examples of communication protocols that may be supported by protocol decoders **30** include the HyperText Transfer Protocol ("HTTP"), the File Transfer Protocol ("FTP"), the Network News Transfer Protocol ("NNTP"), the Simple Mail Transfer Protocol ("SMTP"), Telnet, Domain Name System ("DNS"), Gopher, Finger, the Post Office Protocol ("POP"), the Secure Socket Layer ("SSL") protocol, the Lightweight Directory Access Protocol ("LDAP"), Secure Shell ("SSH"), Server Message Block ("SMB") and other protocols." (Yang, 8:8-48.) *See* also Yang, 10:60-11:10, 11:20-23, 12:55-67, 13:35-41, 14:12-21, 15:29-34, 15:50-62, 15:63-64. *See* claim elements 1[b] and 1[d].
7. The method of claim 1, further comprising:	*See* claim 1.
7[a]. identifying a response corresponding to a request for sharing of data packets, wherein the identified	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
response is not compatible with the received privacy preference; and	For example, Yang discloses determining a blocking response (*e.g.*, dropping packets) in accordance with a determination of a security risk (*e.g.*, a matching pattern). Yang further discloses that the IDS performs its security functions on all packets, including response packets. "As described in further detail below, stateful inspection engine **28** inspects both client-to-server packet flows as well as server-to-client packet flows in order to more accurately identify the type of application and underlying protocol for each communication session. This may assist when, for example, a malicious user attempts to spoof (i.e., mimic) one type of application and instead use another in attempt to bypass an IDS. As an example, a malicious user may attempt to circumvent an IDS by spoofing an SMTP request when actually using the HTTP protocol. IDS **20** may determine from the response from the server that the original packet flow was just an attempt to bypass IDS **20** and may take appropriate action, such as dropping future packets associated with the packet flow and/or alerting the targeted device of the attack. IDS **20** may use a minimum data size of the reassembled TCP segments, in addition to the signature, in order to identify the types of applications. Certain applications may require a minimum amount of data, so IDS **20** may distinguish malicious packet flows by determining whether the packet flow contains enough data for the identified protocol. Moreover, IDS **20** may not necessarily recognize every application. In one embodiment, when an application is unknown, IDS **20** may simply forward the packet flow. If IDS **20** cannot identify a given application, it may be because that application is not a typical target for a malicious packet flow. Other embodiments may take other actions for unidentified applications, however, such as discarding all packets which target unknown applications or applying a default signature to all packet flows associated with unknown application types. Other embodiments may also utilize other protocols, such as the user datagram protocol (UDP); IDS **20** accordingly may require a minimum data size of UDP segments in order to identify the application associated with the UDP segments.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	For each packet flow, stateful inspection engine **28** buffers a copy of the packet flow and reassembles the buffered packet flow to form application-layer communications **32**. For example, stateful inspection engine **28** may reconstruct TCP segments into application-layer communications **32**, which represent protocol-specific messages. Stateful inspection engine **28** invokes the appropriate one of protocol decoders **30** based on the identified type of application determination to analyze the application-layer communications **32**. Protocol decoders **30** represent a set of one or more protocol-specific software modules. Each of protocol decoders **30** corresponds to a different communication protocol or service. Examples of communication protocols that may be supported by protocol decoders **30** include the HyperText Transfer Protocol ("HTTP"), the File Transfer Protocol ("FTP"), the Network News Transfer Protocol ("NNTP"), the Simple Mail Transfer Protocol ("SMTP"), Telnet, Domain Name System ("DNS"), Gopher, Finger, the Post Office Protocol ("POP"), the Secure Socket Layer ("SSL") protocol, the Lightweight Directory Access Protocol ("LDAP"), Secure Shell ("SSH"), Server Message Block ("SMB") and other protocols." (Yang, 7:61-8:48.) "Upon receiving data from protocol decoders **30**, stateful inspection engine **28** selects the attack definitions **33** that are defined for the corresponding protocol, and optionally a set of default attack definitions when no particular type of application has been identified (**88**). Stateful inspection engine **28** then applies the selected compound attack definitions to determine whether the communication session represents a security risk (**90**). When applying a given compound attack definition, stateful inspection engine **28** determines whether all of the specified patterns and protocol anomalies are satisfied for any given communication session between peers, either on a per-transaction basis or over the lifetime of the communication session, as specified by the compound attack definition. Moreover, if required by the compound attack definition, stateful inspection engine **28** may determine whether the specified patterns and protocol anomalies are satisfied in a required order.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	In the event a security risk (i.e., match) is detected (**91**), stateful inspection engine **28** outputs alert **40** to security management module **44** for logging and further analysis (**92**). In addition, stateful inspection engine **28** may take any of a number of programmed responses, such as dropping the packets associated with the communication session, automatically closing the communication session or other action. If no security risk is detected for a given application-layer communication session, forwarding component **31** forwards the packet flow to the destination (**94**). IDS **20** then waits for a response packet flow and reanalyzes the packet flow in light of the response to check the initial determination." (Yang, 12:51-13:12.) *See* claim elements 1[c] and 1[d].
7[b]. providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses maintaining communications between the IDS and the user's device, including blocking and allowing some communications according to whether a security risk is determined. *See* claim elements 1[d] and 1[e].
8. The method of claim 7, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification, blocking, data randomization, content modification, change of encoding, change of file template, change of header, change of	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses blocking communications (*e.g.*, dropping packets) in accordance with a determination of a security risk (*e.g.*, a matching pattern). *See* claim elements 1[c] and 1[d].

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
footer, addition of a predetermined data packet and encryption.	
9. A method of dynamic management of private data during communication between a remote server and a user's device, the method comprising:	*See* claim element 1[pre].
9[a]. receiving, by the user's device, a request for retrieval of at least one data packet from the user's device, wherein the user's device is configured to provide a response corresponding to the received request;	*See* claim element 1[a].
9[b]. determining, by the remote server, at least one communication data pattern corresponding to the received request, wherein the at least one communication data pattern is determined in accordance with a behavior range of the communication data packet, and wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during communication between the remote server and the user's device;	*See* claim elements 1[b] and 6[a], and claim 4.
9[c]. receiving, by the user's device, a privacy preference for the user's	*See* claim element 1[c].

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
device, wherein the privacy preference compromises a list of allowed data patterns for sharing during communication;	
9[d]. modifying data packets corresponding to requests for sharing of responses and corresponding data patterns that are not compatible with the received privacy preference; and	*See* claim elements 1[d] and 6[b], and claim 3.
9[e]. maintaining communication between the remote server and the user's device, with sharing of the modified data packets.	*See* claim element 1[e].
10. The method of claim 9, wherein the communication data pattern of the at least one response is determined from metadata of communication with the remote server.	*See* claim 2.
11. The method of claim 9, further comprising:	*See* claim 9.
11[a]. determining, by the remote server, the communication data type of the at least one data packet of the received request; and	*See* claim element 1[b].
11[b]. modifying data packets corresponding to requests for sharing of responses and corresponding data	*See* claim elements 1[d] and 3[b].

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
patterns, corresponding to at least one communication data packet type, that are not compatible with the received privacy preference,	
11[c]. wherein the privacy preference comprises a list of allowed data packet types for sharing during communication.	*See* claim element 1[c].
12. The method of claim 11, wherein the communication data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, keywords, driver, and communication protocol.	*See* claim 5.
13. The method of claim 9, wherein the user's device is in communication with an external database with at least one communication data pattern corresponding to a request for sharing of data packets from the user's device and wherein at least one data pattern compatible with the received request is determined from the database.	*See* claim elements 1[pre], 1[b], 1[c], and 1[d], and claim 3.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
14. The method of claim 9, further comprising:	*See* claim 9.
14[a]. determining, by the remote server, a type of the at least one data packet of the received request; and	*See* claim element 1[b].
14[b]. linking at least one data pattern to a type of data packet from the user's device, wherein the linking is based on the communication data type and the communication data pattern of the at least one response.	*See* claim element 6[b].
15. The method of claim 9, further comprising:	*See* claim 9.
15[a]. identifying a communication data pattern corresponding to a request for sharing of data packet types, wherein the identified data pattern is not compatible with the received privacy preference; and	*See* claim 3.
15[b]. providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.	*See* claim 3.
16. The method of claim 15, wherein the modification of the at least one modified data packet is selected from the group consisting of data	*See* claim 8.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
nullification, blocking, data randomization, content modification, change of encoding, change of file template, change of header, change of footer, addition of a predetermined data packet and encryption.	
17. A system for dynamic management of private data during communication between a remote server and at least one user's device, the system comprising:	To the extent the preamble is limiting, Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses a system for dynamically managing private network communications between an intrusion detection system (IDS) and a user device (*e.g.*, a "node," a security management device). The communications may include encrypted communications. *See* also claim element 1[pre].
17[a]. a memory;	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang's system comprises a memory. "FIG. 3 is a block diagram that illustrates an example embodiment of stateful inspection engine **28**. In the example embodiment, stateful inspection engine **28** includes reassembly module **50**, application identification module **51**, and attack detection module **52**. In addition, stateful inspection engine **28** includes patterns table **54**, data buffer **55**, anomalies table **56**, and attack definitions **33**. Reassembly module **50** receives inbound network traffic **24** and reassembles application-layer communications **32** from the packet flows. Reassembly module **50** forwards the

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	reassembled application-layer communications **32** to the appropriate protocol decoders **30** for processing. Stateful inspection engine **28** stores attack definitions **33** received from security management module **44**. Attack definitions **33** may be stored, for example, in a computer-readable medium, such as random access memory ("RAM"). Each of attack definitions **33** specifies a combination of one or more patterns specified within patterns table **54** and one or more protocol-specific anomalies specified within anomalies table **56**." (Yang, 9:6-26.)
17[b]. a communication data type database, comprising at least one communication data type corresponding to sharing of at least one data packet from the user's device;	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discloses techniques for specific application and protocols. Yang describes identifying an application or protocol based on a TCP port in a packet header. This technique employs a "static port mapping" that associates exemplary TCP port numbers with exemplary applications and protocols. The mapping can be stored in Yang's memory. "Application identification module **51** identifies the type of application and underlying protocol for each intercepted communication session. When stateful inspection engine **28** receives a packet as part of a packet flow, reassembly module **50** buffers the packet in data buffer **55**. In one embodiment, data buffer **55** may store data as a sliding window. That is, data buffer **55** may store data until becoming full or reaching a specified required amount of minimum data for identification. When full, data buffer **55** discards certain data to make room for storing new data. In one embodiment, data buffer **55** may store and discard data according to a first-in, first-out ("FIFO")-like protocol wherein the first data to be stored is the first data to be discarded when data buffer **55** becomes full. In another embodiment, data buffer **55** may discard data according to a least recently used protocol wherein, when data buffer **55** is full, the packet flow which has been least recently used will be discarded to make room for new data to be stored.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	In one embodiment, reassembly module **50** may associate packets in a packet flow, and packet flows as a communication session, according to the 5-tuple {source IP address, destination IP address, protocol, source port, destination port}. Other embodiments may use other forms of associating packets. For example, in one embodiment, IDS **20** may be part of a network that utilizes virtual local area networks (VLANs). Accordingly, reassembly module **50** may associate packets in a packet flow according to a VLAN identifier, a source address, and a destination address. In any case, reassembly module **50** may utilize the information maintained within flow table **35** (FIG. 2) to reassemble network data, e.g., to form reassembled TCP data. Application identification module **51** analyzes the reassembled data for the packet flow to identify the type of application and protocol associated with the packet flow. If application identification module **51** is unable to identify the type of application and protocol associated with the packet flow, application identification module **51** may use the well-known static port binding as a default application selection. Table 1, below, shows an example static port binding list. Other embodiments may use more, fewer, or different entries in a static port table. Moreover, administrator **42** may configure the static port mapping using security management module **44**." (Yang, 9:27-10:3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	TABLE I PORT / APPLICATION 20 FTP 22 SSH 23 Telnet 25 SMTP 43 WHOIS 53 DNS 67 BOOTP or DHCP 70 Gopher 79 Finger 80 HTTP 109 POP 110 POP3 113 ident/IRC 118 SQL 119 NNTP 194 IRC 443 HTTPS 445 SMB 564 RTSP (Yang, Table 1.) "In the case where application identification module **51** is unable to identify a type of application and protocol for a packet flow, application identification module **51** may use the default static port mapping to determine an application, causing IDS **20** to respond accordingly. In some cases, application identification module **51** may not be able to identify the application and the static port mapping may not have an entry for the requested port number. Various embodiments may treat this situation according to specifications of, for example, a system administrator. For example, in one embodiment, IDS **20** simply

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	forwards packet flows with undetermined application types and protocols that cannot be determined by the static port mapping pass, as an unknown application may indicate that the packet flow is not targeting any type of application known to pose a security threat. In other embodiments, IDS **20** may automatically discard packet flows with unknown application types and protocols that cannot be determined by the static port mapping." (Yang, 10:25-42.) *See* also claim element 1[b] and claim 2.
17[c]. a privacy preference database, comprising a list of allowed types of data packets for sharing during communication with the at least one user's device;	*See* claim element 1[c].
17[d]. a communication module, to allow communication between the remote server and the at least one user's device;	*See* claim elements 1[pre] and 1[b].
17[e]. and a processor, coupled to a response database and to the privacy preference database, wherein the processor is configured to instruct the remote server to determine at least one data type for sharing of data packet that is compatible with the list of allowed patterns of data packets for sharing, and wherein the at least one data type is determined in accordance with characteristics of the communication data packet, and	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang's system a processor coupled to a memory for storing request responses and privacy preferences. Yang describes categorizing a packet as pertaining to an "unknown" application or protocol. Yang further discloses a static port binding list. Yang further identifies a variety of applications and protocols that were well-known not to be encrypted, such that packets operating according to these applications and protocols are not read. Burns also refers to other well-known protocols that were known to employ encrypted packets. Yang's technique requires first identifying that an IP packet contains a TCP packet via the IP packet header's "protocol" field. Yang's technique identifies an

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during communication between the remote server and the user's device.	application or protocol via the source and destination port numbers in the TCP packer's header. Yang's IDS "analyze[s] the traffic in real-time prior to forwarding." "In general, IDS **10** identifies packet flows in the monitored traffic, and transparently reassembles application-layer communications from the packet flows. A set of protocol-specific decoders within the IDS **10** analyzes the application-layer communications and identifies application-layer transactions. In general, a "transaction" refers to a bounded series of related application-layer communications between peer devices. For example, a single TCP connection can be used to send (receive) multiple HyperText Transfer Protocol (HTTP) requests (responses). As one example, a single web-page comprising multiple images and links to HTML pages may be fetched using a single TCP connection. An HTTP decoder identifies each request/response within the TCP connection as a different transaction. This may be useful to prevent certain attack definitions from being applied across transaction boundaries. In one embodiment, a transaction may be identified according to source and destination IP address, protocol, and source and destination port numbers. Other embodiments may identify a transaction in other ways, for example, by using media access control ("MAC") addresses." (Yang, 5:1-20.) "In addition, IDS 10 monitors for traffic flowing in the opposite direction to intercept any response from the server. That is, IDS 10 identifies any response packet flow from the server, which typically utilizes the same or substantially similar network addresses, ports and/or protocol as the initial client request. Upon intercepting the response packet flow, IDS 10 performs an additional application identification test to confirm whether its initial assessment of the type of software application and protocol for the particular client-server communication session was correct. If not, IDS 10 may reclassify the communication as a different type of software application and protocol, and select and apply additional attack definitions based on the updated classification. That is, IDS 10 may retrieve the cached assessment of the application and then use the response packet flow to either confirm or modify the assessment, again caching the result." (Yang, 6:22-37.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	"These techniques may allow IDS **10** to provide improved attack detection and prevention in environments where some applications no longer use a static port mapping but instead use dynamic or randomized port assignments in an attempt to evade detection and containment. For example, instant messaging programs may initiate communication on one port, but then agree to communicate thereafter on a distinct port. Other programs, such as peer-to-peer ("P2P") programs and hacker toolkits such as Metasploit, may determine ports dynamically as well. As described, IDS **10** may make an initial determination for the application type of a packet flow from a client device of public network **6** to a server device, e.g. node **8**A, based upon characteristics of the communication in addition to specified port number. IDS **10** may then analyze the return communication from node **8**A to the client device to verify the initial assumption or to modify the initial determination if necessary. In some embodiments, enterprise network **5** may include multiple IDSs **10** and **14** located within different regions (e.g., sub-networks) of enterprise network **5**. Security management device **18** may operate as a central device for managing IDSs **10** and **14**. Although the example illustrated in FIG. 1 is described in terms of dedicated IDSs **10** and **14**, the functionality described herein may be incorporated within other devices, such as firewall **9** or switch **19**." (Yang, 6:45-7:2.) "IDS **20** may use a minimum data size of the reassembled TCP segments, in addition to the signature, in order to identify the types of applications. Certain applications may require a minimum amount of data, so IDS **20** may distinguish malicious packet flows by determining whether the packet flow contains enough data for the identified protocol. Moreover, IDS **20** may not necessarily recognize every application. In one embodiment, when an application is unknown, IDS **20** may simply forward the packet flow. If IDS **20** cannot identify a given application, it may be because that application is not a typical target for a malicious packet flow. Other embodiments may take other actions for unidentified applications, however, such as discarding all packets which target unknown applications or applying a default signature to all packet flows associated with unknown

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	application types. Other embodiments may also utilize other protocols, such as the user datagram protocol (UDP); IDS **20** accordingly may require a minimum data size of UDP segments in order to identify the application associated with the UDP segments. For each packet flow, stateful inspection engine **28** buffers a copy of the packet flow and reassembles the buffered packet flow to form application-layer communications **32**. For example, stateful inspection engine **28** may reconstruct TCP segments into application-layer communications **32**, which represent protocol-specific messages. Stateful inspection engine **28** invokes the appropriate one of protocol decoders **30** based on the identified type of application determination to analyze the application-layer communications **32**. Protocol decoders **30** represent a set of one or more protocol-specific software modules. Each of protocol decoders **30** corresponds to a different communication protocol or service. Examples of communication protocols that may be supported by protocol decoders **30** include the HyperText Transfer Protocol ("HTTP"), the File Transfer Protocol ("FTP"), the Network News Transfer Protocol ("NNTP"), the Simple Mail Transfer Protocol ("SMTP"), Telnet, Domain Name System ("DNS"), Gopher, Finger, the Post Office Protocol ("POP"), the Secure Socket Layer ("SSL") protocol, the Lightweight Directory Access Protocol ("LDAP"), Secure Shell ("SSH"), Server Message Block ("SMB") and other protocols." (Yang, 8:8-48.) "Application identification module **51** identifies the type of application and underlying protocol for each intercepted communication session. When stateful inspection engine **28** receives a packet as part of a packet flow, reassembly module **50** buffers the packet in data buffer **55**. In one embodiment, data buffer **55** may store data as a sliding window. That is, data buffer **55** may store data until becoming full or reaching a specified required amount of minimum data for identification. When full, data buffer **55** discards certain data to make room for storing new data. In one embodiment, data buffer **55** may store and discard data according to a first-in, first-out ("FIFO")-like protocol wherein the first data to be stored is the first data to be discarded when data buffer **55** becomes full. In

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	another embodiment, data buffer **55** may discard data according to a least recently used protocol wherein, when data buffer **55** is full, the packet flow which has been least recently used will be discarded to make room for new data to be stored. In one embodiment, reassembly module **50** may associate packets in a packet flow, and packet flows as a communication session, according to the 5-tuple {source IP address, destination IP address, protocol, source port, destination port}. Other embodiments may use other forms of associating packets. For example, in one embodiment, IDS **20** may be part of a network that utilizes virtual local area networks (VLANs). Accordingly, reassembly module **50** may associate packets in a packet flow according to a VLAN identifier, a source address, and a destination address. In any case, reassembly module **50** may utilize the information maintained within flow table **35** (FIG. 2) to reassemble network data, e.g., to form reassembled TCP data. Application identification module **51** analyzes the reassembled data for the packet flow to identify the type of application and protocol associated with the packet flow. If application identification module **51** is unable to identify the type of application and protocol associated with the packet flow, application identification module **51** may use the well-known static port binding as a default application selection. Table 1, below, shows an example static port binding list. Other embodiments may use more, fewer, or different entries in a static port table. Moreover, administrator **42** may configure the static port mapping using security management module **44**." (Yang, 9:27-10:3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	TABLE I PORT / APPLICATION 20 FTP 22 SSH 23 Telnet 25 SMTP 43 WHOIS 53 DNS 67 BOOTP or DHCP 70 Gopher 79 Finger 80 HTTP 109 POP 110 POP3 113 ident/IRC 118 SQL 119 NNTP 194 IRC 443 HTTPS 445 SMB 564 RTSP (Yang, Table 1.) "In the case where application identification module **51** is unable to identify a type of application and protocol for a packet flow, application identification module **51** may use the default static port mapping to determine an application, causing IDS **20** to respond accordingly. In some cases, application identification module **51** may not be able to identify the application and the static port mapping may not have an entry for the requested port number. Various embodiments may treat this situation according to specifications of, for example, a system administrator. For example, in one embodiment, IDS **20** simply

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	forwards packet flows with undetermined application types and protocols that cannot be determined by the static port mapping pass, as an unknown application may indicate that the packet flow is not targeting any type of application known to pose a security threat. In other embodiments, IDS **20** may automatically discard packet flows with unknown application types and protocols that cannot be determined by the static port mapping." (Yang, 10:25-42.) "Once configured, IDS **20** monitors network traffic **24** (**72**). In some configurations, stateful inspection engine **28** of forwarding plane **22** may receive network traffic and mirror the network traffic for purposes of analysis. Forwarding component **31** seamlessly forwards the original network traffic. In other embodiments, traffic is not mirrored, rather a line-rate buffering approach is used to analyze the traffic in real-time prior to forwarding." (Yang, 11:63-12:3.) *See* also Yang, 2:58-3:17, 9:19-26, 19:36-20:60. *See* also claim 1.
18. The system of claim 17, further comprising a communication data pattern database, coupled to the processor and comprising at least one data pattern corresponding to sharing of at least one data packet from the user's device, wherein the processor is configured to modify data packet corresponding to requests for retrieval of data packets and communication data types that are not compatible with communication data patterns from a communication data pattern database.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang provides details about "protocol-specific anomaly analysis," including using anomaly information "to detect sophisticated attack behaviors." "In the event a security risk is detected, stateful inspection engine **28** outputs alert **40** to security management module **44** (FIG. 2) for logging and further analysis. Stateful inspection engine **28** may also direct forwarding component **31** to automatically drop the packet flow associated with the application-layer communications within which the network attack was detected. In this manner, stateful inspection engine **28** combines pattern matching with protocol-specific anomaly analysis to detect sophisticated attack behaviors." (Yang, 11:43-51.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	See also claim elements 1[b] and 1[d], and claim 3.
19. The system of claim 18, wherein data packets from the user's device are selected from the group consisting of user's device files, user's device characteristics, user's device indirect attributes, user's device sensor data, user's device browser data, user's device form data, user's device dynamic memory and user's device static memory.	Yang discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Yang discusses analyzing HTTP traffic, which was used for communicating "web-page" data to and from a web browser. Yang also describes creating a compound attack definition (a filter) that applies only to "HTTP form data." "In general, IDS **10** identifies packet flows in the monitored traffic, and transparently reassembles application-layer communications from the packet flows. A set of protocol-specific decoders within the IDS **10** analyzes the application-layer communications and identifies application-layer transactions. In general, a "transaction" refers to a bounded series of related application-layer communications between peer devices. For example, a single TCP connection can be used to send (receive) multiple HyperText Transfer Protocol (HTTP) requests (responses). As one example, a single web-page comprising multiple images and links to HTML pages may be fetched using a single TCP connection. An HTTP decoder identifies each request/response within the TCP connection as a different transaction. This may be useful to prevent certain attack definitions from being applied across transaction boundaries. In one embodiment, a transaction may be identified according to source and destination IP address, protocol, and source and destination port numbers. Other embodiments may identify a transaction in other ways, for example, by using media access control ("MAC") addresses." (Yang, 5:1-20.) "One example of such a protocol event is repeated failure of an FTP login request. Example anomalies for the HTTP protocol include missing HTTP version information, malformed universal resource locators ("URLs"), directory traversals, header overflow, authentication overflow and cookie overflow. Example anomalies for SMTP protocol include too many recipients, relay attempts, and domain names that exceed a defined length. Example anomalies for the POP3 protocol include user overflow and failed logins.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	Example anomalies for the FTP protocol include missing arguments, usernames or pathnames that exceed a defined length and failed logins. Other anomalies include abnormal and out-of-specification data transmissions, and commands directing devices to open network connections to devices other than the client devices issuing the commands." (Yang, 5:3-52.) "FIG. 6 is an example user interface **100** presented by an IDS with which an administrator interacts to create a pattern for detecting attacks. In this example, user interface **100** allows the administrator to define a compound network attack definition. As illustrated in FIG. 6, user interface **100** includes an input **102** by which the administrator may provide a name for the compound network attack. In addition, interface **100** includes an input **104** by which the administrator may provide a brief description for the compound network attack. The example of FIG. 6 illustrates a compound network attack for detecting directory traversal attempts against an Apache web server. In addition, interface **100** includes inputs **106** by which the administrator may provide other data for the compound network attack, such as a severity level **108**, keywords for use in indexing and analysis, and other data. FIG. 7 is an example of user interface **110** presented when the user selects the "Signatures" tab of user interface **100** (FIG. 6). As illustrated, user interface **110** includes an input **112** with which the administrator specifies a protocol to which the compound attack definition pertains. In this example, the administrator has selected HTTP as this exemplary compound network attack definition relates to an Apache web server. User interface **110** further includes an input **111** by which the administrator selects from a list of contexts for the particular protocol selected in input **112**. For example, input **111** provides a drop-down menu by which the administrator selects HTTP contexts, such as HTTP Authorization, HTTP data, HTTP form data and other contexts available from the headers in an HTTP communication.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Yang**
	User interface **110** further includes an input **114** by which the administrator selects a scope for the compound attack definition is to be applied. In particular, input **114** allows the administrator to specify whether the attack is session based (i.e., to be applied over the life-time of a session and across all transactions associated with the session) or transaction based (i.e., applied on a per transaction basis and cleared at the end of each transaction). User interface **110** further includes an input area **118** by which the administrator defines the constituent members of the compound attack definition. In particular, the administrator may add patterns by selecting Add button **120** or protocol anomalies by selecting Add button **122**. Input area **118** lists the members currently defined for the compound attack definition. In this example, two patterns **124**, **126** have been defined in the form of regular expressions. User interface **110** also includes an input **130** by which the administrator controls whether the constituent members must be detected in the order presented within input area **118** or whether the members may be detected in any order within the scope defined by input area **114**." (Yang, 15:11-62.) *See* also claim elements 1[pre] and 1[b].
20. The system of claim 18, wherein at least the communication module and the processor are embedded on a single hardware component.	*See* claim element 1[pre] and claim 17.