SYSTEMS AND METHODS FOR ENHANCING ELECTRONIC COMMUNICATION SECURITY

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS

This application claims priority to U.S. patent application nos. 10/361,091 and 10/361,067 filed on February 7, 2003 and application nos. 10/093,553, 10/094,211, and 10/094,266 all filed on March 8, 2002, which applications are all hereby incorporated herein in their entirety.

BACKGROUND

The present invention is directed to systems and methods for enhancing security associated with electronic communications. More specifically, without limitation, the present invention relates to computer-based systems and methods for assessing security risks associated with electronic communications transmitted over a communications network. Further, the present invention in some embodiments relates to computer-based systems and methods for assessing security risks associated with electronic communications transmitted over a communications network and for responding to a range of threats to messaging systems.

The Internet is a global network of connected computer networks. Over the last several years, the Internet has grown in significant measure. A large number of computers on the Internet provide information in various forms. Anyone with a computer connected to the Internet can potentially tap into this vast pool of information.

The information available via the Internet encompasses information available via a variety of types of application layer information servers such as SMTP (simple mail transfer protocol), POP3 (Post Office Protocol), GOPHER (RFC 1436), WAIS, HTTP (Hypertext Transfer Protocol, RFC 2616) and FTP (file transfer protocol, RFC 1123).

One of the most wide spread method of providing information over the Internet is via the World Wide Web (the Web). The Web consists of a subset of the computers connected to the Internet; the computers in this subset run Hypertext Transfer Protocol (HTTP) servers (Web servers). Several extensions and modifications to HTTP have

been proposed including, for example, an extension framework (RFC 2774) and authentication (RFC 2617). Information on the Internet can be accessed through the use of a Uniform Resource Identifier (URI, RFC 2396). A URI uniquely specifies the location of a particular piece of information on the Internet. A URI will typically be composed of several components. The first component typically designates the protocol by which the address piece of information is accessed (e.g., HTTP, GOPHER, etc.). This first component is separated from the remainder of the URI by a colon (':'). The remainder of the URI will depend upon the protocol component. Typically, the remainder designates a computer on the Internet by name, or by IP number, as well as a more specific designation of the location of the resource on the designated computer. For instance, a typical URI for an HTTP resource might be:

http://www.server.com/dir1/dir2/resource.htm

where http is the protocol, www.server.com is the designated computer and /dir1/dir2/resouce.htm designates the location of the resource on the designated computer. The term URI includes Uniform Resource Names (URN's) including URN's as defined according to RFC 2141.

Web servers host information in the form of Web pages; collectively the server and the information hosted are referred to as a Web site. A significant number of Web pages are encoded using the Hypertext Markup Language (HTML) although other encodings using eXtensible Markup Language (XML) or XHTML. The published specifications for these languages are incorporated by reference herein; such specifications are available from the World Wide Web Consortium and its Web site (http://www.w3c.org). Web pages in these formatting languages may include links to other Web pages on the same Web site or another. As will be known to those skilled in the art, Web pages may be generated dynamically by a server by integrating a variety of elements into a formatted page prior to transmission to a Web client. Web servers, and information servers of other types, await requests for the information from Internet clients.

Client software has evolved that allows users of computers connected to the Internet to access this information. Advanced clients such as Netscape's Navigator and

Microsoft's Internet Explorer allow users to access software provided via a variety of information servers in a unified client environment. Typically, such client software is referred to as browser software.

Electronic mail (e-mail) is another wide spread application using the Internet. A variety of protocols are often used for e-mail transmission, delivery and processing including SMTP and POP3/MAPI (Messaging Application Programming Interface) as discussed above. These protocols refer, respectively, to standards for communicating e-mail messages between servers and for server-client communication related to e-mail messages. These protocols are defined respectively in particular RFC's (Request for Comments) promulgated by the IETF (Internet Engineering Task Force). The SMTP protocol is defined in RFC 821 and 822, and the POP3 protocol is defined in RFC 1939. MAPI is a protocol developed by Microsoft (Microsoft Corp.; Redmond, WA) for allowing higher level communication and organization for mail-capable application than provided through the POP3 protocol; the reference manual for MAPI can be found through Microsoft's online reference manual (http://msdn.microsoft.com/library/). In addition, the IMAP protocol has evolved as an alternative to POP3 that supports more advanced interactions between e-mail servers and clients. This protocol is described in RFC 2060.

Since the inception of these standards, various needs have evolved in the field of e-mail leading to the development of further standards including enhancements or additional protocols. For instance, various enhancements have evolved to the SMTP standards leading to the evolution of extended SMTP. Examples of extensions may be seen in (1) RFC 1869 that defines a framework for extending the SMTP service by defining a means whereby a server SMTP can inform a client SMTP as to the service extensions it supports and in (2) RFC 1891 that defines an extension to the SMTP service, which allows an SMTP client to specify (a) that delivery status notifications (DSNs) should be generated under certain conditions, (b) whether such notifications should return the contents of the message, and (c) additional information, to be returned with a DSN, that allows the sender to identify both the recipient(s) for which the DSN was issued, and the transaction in which the original message was sent.

Both HTTP and SMTP communicate in a standard configuration communicate messages over an open (unencrypted) channel. To enhance security of transmissions various technological advance have been implemented. For both these protocols, two approaches have evolved: channel encryption and message encryption. Channel encryption provides for establishing a secure channel where any amount of data can be communicated using an established encryption. Message encryption provides for establishing encryption of individual messages, which are then forwarded over a particular channel. These approaches can be combined leading to two levels of encryption: one at the channel level and one at the message level.

For HTTP, a particular form of message level encryption has been adopted as a standard referred to as S-HTTP. The specifics of this protocol can be found in RFC 2660. HTTP requests and responses support communication of data according to the MIME (Multipurpose Internet Mail Extensions) standard (RFC). A security enhanced version of this has been implemented and referred to as S/MIME (Secure/ Multipurpose Internet Mail Extensions); this security enhanced version also can provide for a measure of message level encryption. Specific details of S/MIME can be found at RFC 2311 and 2633; additional details surrounding use of S/MIME are defined in a variety of other RFC's including without limitation RFC 2312, 2632, 2634, 2785-6, 2984, 3058, 3114, 3125-6, 3183, 3185, 3211, 3217-8, 3274, 3278, 3369-70 and 3394 (further details on current S/MIME development can be found at the IETF's S/MIME Charter home page http://www.ietf.org/html.charters/smime-charter.html).

For channel level encryption, early development by Netscape Communications led to the SSL (Secure Socket Layer) protocol; documentation for version 3.0 can be found at http://wp.netscape.com/eng/ssl3/ssl-toc.html. This channel encryption mechanism is commonly used and URLs indicate use of this protocol through use of https: rather than http:. A newer technology that is intended as backward compatible with SSL is TLS (Transport Layer Security); a full description of this can be found in RFC 2246. Additional details surrounding use of TLS are defined in a variety of other RFC's including without limitation RFC 2712, 2817-8 and 3268 (further details on

current TLS development can be found at the IETF's TLS Charter home page http://www.ietf.org/html.charters/smime-charter.html).

For SMTP, similar technologies have been applied. For message level encryption, various forms of public key encryption technology have been used. One of the most prevalently used public key encryption technologies is referred to as PGP (PRETTY GOOD PRIVACY). S/MIME can also be used in conjunction with SMTP delivered messages. As with HTTP, SSL or TLS can be used as a channel level encryption mechanism. Further, both channel and one or more forms of message encryption can be used in connection with secure SMTP delivery.

The various standards discussed above are hereby incorporated by reference herein for all purposes. The standards referred to by RFC's are available to the public through the IETF and can be retrieved from its Web site (http://www.ietf.org/rfc.html). The specified protocols are not intended to be limited to the specific RFC's quoted herein above but are intended to include extensions and revisions thereto. Such extensions and/or revisions may or may not be encompassed by current and/or future RFC's.

A host of e-mail server and client products have been developed in order to foster e-mail communication over the Internet. E-mail server software includes such products as sendmail-based servers, Microsoft Exchange, Lotus Notes Server, and Novell GroupWise; sendmail-based servers refer to a number of variations of servers originally based upon the sendmail program developed for the UNIX operating systems. A large number of e-mail clients have also been developed that allow a user to retrieve and view e-mail messages from a server; example products include Microsoft Outlook, Microsoft Outlook Express, Netscape Messenger, and Eudora. In addition, some e-mail servers, or e-mail servers in conjunction with a Web server, allow a Web browser to act as an e-mail client using the HTTP standard.

As the Internet has become more widely used, it has also created new risks for corporations. Breaches of computer security by hackers and intruders and the potential for compromising sensitive corporate information are a very real and serious threat.

Organizations have deployed some or all of the following security technologies to protect their networks from Internet attacks:

Firewalls have been deployed at the perimeter of corporate networks. Firewalls act as gatekeepers and allow only authorized users to access a company network. Firewalls play an important role in controlling traffic into networks and are an important first step to provide Internet security.

Intrusion detection systems (IDS) are being deployed throughout corporate networks. While the firewall acts as a gatekeeper, IDS act like a video camera. IDS monitor network traffic for suspicious patterns of activity, and issue alerts when that activity is detected. IDS proactively monitor your network 24 hours a day in order to identify intruders within a corporate or other local network.

Firewall and IDS technologies have helped corporations to protect their networks and defend their corporate information assets. However, as use of these devices has become widespread, hackers have adapted and are now shifting their point-of-attack from the network to Internet applications. The most vulnerable applications are those that require a direct, "always-open" connection with the Internet such as web and e-mail. As a result, intruders are launching sophisticated attacks that target security holes within these applications.

Many corporations have installed a network firewall, as one measure in controlling the flow of traffic in and out of corporate computer networks, but when it comes to Internet application communications such as e-mail messages and Web requests and responses, corporations often allow employees to send and receive from or to anyone or anywhere inside or outside the company. This is done by opening a port, or hole in their firewall (typically, port 25 for e-mail and port 80 for Web), to allow the flow of traffic. Firewalls do not scrutinize traffic flowing through this port. This is similar to deploying a security guard at a company's entrance but allowing anyone who looks like a serviceman to enter the building. An intruder can pretend to be a serviceman, bypass the perimeter security, and compromise the serviced Internet application.

FIG. 1 depicts a typical prior art server access architecture. With in a corporation's local network 190, a variety of computer systems may reside. These systems typically include application servers 120 such as Web servers and e-mail servers, user workstations running local clients 130 such as e-mail readers and Web browsers, and data storage devices 110 such as databases and network connected disks. These systems communicate with each other via a local communication network such as Ethernet 150. Firewall system 140 resides between the local communication network and Internet 160. Connected to the Internet 160 are a host of external servers 170 and external clients 180.

Local clients 130 can access application servers 120 and shared data storage 110 via the local communication network. External clients 180 can access external application servers 170 via the Internet 160. In instances where a local server 120 or a local client 130 requires access to an external server 170 or where an external client 180 or an external server 170 requires access to a local server 120, electronic communications in the appropriate protocol for a given application server flow through "always open" ports of firewall system 140.

The security risks do not stop there. After taking over the mail server, it is relatively easy for the intruder to use it as a launch pad to compromise other business servers and steal critical business information. This information may include financial data, sales projections, customer pipelines, contract negotiations, legal matters, and operational documents. This kind of hacker attack on servers can cause immeasurable and irreparable losses to a business.

In the 1980's, viruses were spread mainly by floppy diskettes. In today's interconnected world, applications such as e-mail serve as a transport for easily and widely spreading viruses. Viruses such as "I Love You" use the technique exploited by distributed Denial of Service (DDoS) attackers to mass propagate. Once the "I Love You" virus is received, the recipient's Microsoft Outlook sends emails carrying viruses to everyone in the Outlook address book. The "I Love You" virus infected millions of computers within a short time of its release. Trojan horses, such as Code Red use this

same technique to propagate themselves. Viruses and Trojan horses can cause significant lost productivity due to down time and the loss of crucial data.

The Nimda worm simultaneously attacked both email and web applications. It propagated itself by creating and sending infectious email messages, infecting computers over the network and striking vulnerable Microsoft IIS Web servers, deployed on Exchange mail servers to provide web mail.

Most e-mail and Web requests and responses are sent in plain text today, making it just as exposed as a postcard. This includes the e-mail message, its header, and its attachments, or in a Web context, a user name and password and/or cookie information in an HTTP request. In addition, when you dial into an Internet Service Provider (ISP) to send or receive e-mail messages, the user ID and password are also sent in plain text, which can be snooped, copied, or altered. This can be done without leaving a trace, making it impossible to know whether a message has been compromised.

As the Internet has become more widely used, it has also created new troubles for users. In particular, the amount of "spam" received by individual users has increased dramatically in the recent past. Spam, as used in this specification, refers to any communication receipt of which is either unsolicited or not desired by its recipient.

The following are additional security risks caused by Internet applications:

- E-mail spamming consumes corporate resources and impacts productivity. Furthermore, spammers use a corporation's own mail servers for unauthorized email relay, making it appear as if the message is coming from that corporation.
- E-mail and Web abuse, such as sending and receiving inappropriate messages and Web pages, are creating liabilities for corporations. Corporations are increasingly facing litigation for sexual harassment or slander due to e-mail their employees have sent or received.
- Regulatory requirements such as the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act (regulating financial institutions) create liabilities for companies where confidential patient

or client information may be exposed in e-mail and/or Web servers or communications including e-mails, Web pages and HTTP requests.

Using the "always open" port, a hacker can easily reach an appropriate Internet application server, exploit its vulnerabilities, and take over the server. This provides hackers easy access to information available to the server, often including sensitive and confidential information. The systems and methods according to the present invention provide enhanced security for communications involved with such Internet applications requiring an "always-open" connection.

Anti-spam systems in use today include fail-open systems in which all incoming messages are filtered for spam. In these systems, a message is considered not to be spam until some form of examination proves otherwise. A message is determined to be spam based on an identification technique. Operators of such systems continue to invest significant resources in efforts to reduce the number of legitimate messages that are misclassified as spam. The penalties for any misclassification are significant and therefore most systems are designed to be predisposed not to classify messages as spam.

One such approach requires a user to explicitly list users from whom email is desirable. Such a list is one type of "whitelist". There are currently two approaches for creating such a whitelist. In a desktop environment, an end-user can import an address book as the whitelist. This approach can become a burden when operated at a more central location such as the gateway of an organization. Therefore, some organizations only add a few entries to the whitelist as necessary. In that case, however, the full effect of whitelisting is not achieved. The present invention improves upon these systems by including a system that allows a more effective solution for whitelisting while requiring reduced manual effort by end-users or administrators. The present invention also allows a whitelist system to be strengthened by authenticating sender information.

Other systems in use today employ a fail-closed system in which a sender must prove its legitimacy. A common example of this type of system uses a challenge and response. Such a system blocks all messages from unknown senders and itself sends a

confirmation message to the sender. The sender must respond to verify that it is a legitimate sender. If the sender responds, the sender is added to the whitelist. However, spammers can create tools to respond to the confirmation messages. Some confirmation messages are more advanced in an effort to require that a human send the response. The present invention is an improvement upon these systems. The present invention can reference information provided by users to determine who should be whitelisted rather than rely on the sender's confirmation. The systems and methods according to the present invention provide enhanced accuracy in the automated processing of electronic communications.

U.S. Patent No. 6,052,709, the disclosure of which is incorporated herein by this reference, assigned to Bright Light Technologies discloses a system for collecting spam messages so that rules can be created and sent to servers. The disclosed system includes the steps of data collection, rule creation, and distribution of rules to clients. The disclosed system is directed to a particular method of data collection for spam messages. No system or method for creating rules based on input data are disclosed. Nor does it disclose a systematic approach to generating rules. Furthermore, the disclosed system is limited to spam threats and only allows one type of input. The threat management center of the present invention is operative on all messaging threats including, but not limited to, spam, virus, worms, Trojans, intrusion attempts, etc. The threat management center of the present invention also includes novel approaches to the process of rule creation. Additionally, the present invention improves on the state of the art by providing a more generalized and useful data collection approach. The data collection system of the present invention includes modules that process input into data that can be used by the rule creation process. The present invention can also use feedback from application layer security servers as input to the rule creation process.

U.S. Patent Application Serial No. 10/154,137 (publication 2002/0199095 A1), the disclosure of which is incorporated herein by this reference, discloses a system for message filtering. The disclosed system allows spam messages to be forwarded to a database by users of the system. In contrast, the systems and methods of the present invention do not rely on the users; rather the messaging security system(s) can

automatically determine spam using identification techniques and then forward the results to a database. The system of the present invention can add known spam messages as well as misclassified messages forwarded by users to the database to retrain the system. Systems known in the art require the forwarding of entire messages to the databases. In the present invention, individual messaging or application layer security systems can extract meaningful features from spam messages, threatening messages and/or non-spam/non-threatening messages and forward only relevant features to a database.

U.S. Patent No. 6,161,130, the disclosure of which is incorporated herein by this reference, discloses a technique for detecting "junk" email. The disclosed system is operative only on spam and not the entire class of messaging security threats. The inputs for the disclosed system are limited spam and non-spam e-mail. This patent discloses text analysis based features such as the tokens in a message. This patent discloses "predefined handcrafted distinctions" but does not further disclose what they are or how these can be created. The system of the present invention can classify based on not only the text analysis but also other features of messages. Additionally, the system of the present invention can include fully automated feature extraction for non-text based features.

In addition, known security systems have been developed to provide peer-to-peer communication of threat information. Such systems are typically designed for a ring of untrusted peers and therefore address trust management between the peers. Additionally, current peer-to-peer systems do not have a central entity. The system of the present invention operates between a set of trusted peers; therefore, trust management need not be addressed by the present invention. Further, a centralized threat management system coordinates threat information among multiple trusted application layer security systems communicating in a peer-to-peer manner. Therefore, the threat notification system can process more real-time data exchange. This makes the distributed IDS (intrusion detection system) more scalable.

In addition, current systems only exchange intrusion alerts. These systems can only notify each other of attacks of which they are aware. While the underlying

detection method could be misuse or anomaly detection, the data exchanged is only the detected attack information. The system of the present invention distributes more general information about traffic patterns as well as specific threat information. As a non-limiting example, if anomaly detection is used, the system of the present invention can exchange the underlying statistics instead of waiting for the statistics to indicate an attack. Exchanged statistics can include information about the frequency of certain attacks. Therefore, even if other systems already have a signature for a certain attack, the system of the present invention will notify them of an outbreak of this attack. Additionally, traffic patterns can be exchanged among peers and that information can be further processed by the other peers to infer a global view of traffic patterns. This information exchange can be similar to routing protocols that allow each node to infer a global view of the network topology.

SUMMARY

The present invention is directed to systems and methods for secure delivery of electronic communications. A typical architecture can include one or more of the following components: 1) a centralized threat management center that can collect threat information and create rules and/or policies for messaging security systems, 2) a peer-to-peer based messaging notification system that is operative between messaging security systems, and 3) a hierarchical messaging pushback system that blocks communications as close as possible to the source by sending notifications to systems on a path towards the source.

A preferred embodiment according to the present invention for a secure electronic communication delivery system, or as an overall environment supporting a variety of features including secure delivery, includes a system data store (SDS), a system processor and one or more interfaces to one or more communications networks over which electronic communications are transmitted and received. The SDS stores data needed to provide the desired system functionality and may include, for example, received communications, data associated with such communications, information related to known security risks, configuration data regarding secure delivery mechanisms, recipient secure delivery preferences, information related to corporate policy with respect to communications for one or more applications (e.g., corporate e-mail policy, Web access guidelines, message interrogation parameters, and whitelists) and predetermined responses to the identification of particular security risks, situations or anomalies.

The SDS may include multiple physical and/or logical data stores for storing the various types of information. Data storage and retrieval functionality may be provided by either the system processor or data storage processors associated with the data store. The system processor is in communication with the SDS via any suitable communication channel(s); the system processor is in communication with the one or more interfaces via the same, or differing, communication channel(s). The system processor may include one or more processing elements that provide electronic

communication reception, transmission, interrogation, analysis and/or other functionality.

In a threat management center, the SDS may further include one or more sets of threat management goals and/or one or more sets of test data. Accordingly, one preferred threat management method includes a variety of steps that may, in certain embodiments, be executed by the environment summarized above and more fully described below or be stored as computer executable instructions in and/or on any suitable combination of computer-readable media. Threat information is received from one or more sources; such sources can include external security databases and threat information data from one or more application and/or network layer security systems. The received threat information is reduced into a canonical form. Features are extracted from the reduced threat information; these features in conjunction with configuration data such as goals are used to produce rules. In some embodiments, these rules are tested against one or more sets of test data and compared against the same or different goals; if one or more tests fail, the rules are refined until the tests succeed within an acceptable margin of error. The rules are then propagated to one or more application layer security systems.

One preferred threat pushback method includes a variety of steps that may, in certain embodiments, be executed by the environment summarized above and more fully described below or be stored as computer executable instructions in and/or on any suitable combination of computer-readable media. A communication is received. A threat profile associated with the received communication is generated. In some cases, the generation occurs through application of one or more tests to the received communication, wherein each of the one or more tests evaluates the received communication for a particular security risk. In other instance, a manual entry of a threat profile via a provided interface serves to generate the threat profile. The threat profile is compared with configuration information. Typically, configuration information can include threat types of interest and weights associated therewith. In some embodiments, the comparison is accomplished by calculating a threat value from the threat profile and determining whether the threat value satisfies a predetermined

threat condition. If the comparison indicates the received communication represents a threat, one or more computer addresses in a back path of the received communication are identified, and information based upon the stored threat profile is outputted.

In some embodiments, identified address along the back path are authenticated prior to propagation of threat information. In other embodiments, an interface may be provided to allow establishing configuration information regarding one or more threat types, wherein configuration information comprises threat types of interest and weights associated therewith.

Accordingly, one preferred method of whitelist usage includes a variety of steps that may, in certain embodiments, be executed by the environment summarized above and more fully described below or be stored as computer executable instructions in and/or on any suitable combination of computer-readable media. In some embodiments, an electronic communication directed to or originating from an application server is received. The source of the electronic communication may be any appropriate internal or external client or any appropriate internal or external application server. One or more tests are applied to the received electronic communication to evaluate the received electronic communication for a particular security risk. A risk profile associated with the received electronic communication is stored based upon this testing. The stored risk profile is compared against data accumulated from previously received electronic communications to determine whether the received electronic communication is anomalous. If the received communication is determined to be anomalous, an anomaly indicator signal is output. The output anomaly indicator signal may, in some embodiments, notify an application server administrator of the detected anomaly by an appropriate notification mechanism (e.g., pager, e-mail, etc.) or trigger some corrective measure such as shutting down the application server totally, or partially (e.g., deny access to all communications from a particular source).

Some embodiments may provide support for communicating information based upon the stored risk profile to a threat notification system to a further security appliance or further security appliances. Without limitation, such security appliances can include threat management centers and other application layer security systems. Such

communication of information can be instead of, or in addition to, any anomaly indicator signal. In some embodiments, anomaly detection need not occur nor does an anomaly indicator signal need to be output.

In some embodiments, an electronic communication directed to or originating from an email server is received. One or more tests can be applied to the received electronic communication to compare the sender's address in the received electronic communication to addresses contained in one or more whitelists.

Some embodiments may also support a particular approach to testing the received electronic communication, which may also be applicable for use in network level security and intrusion detection. In such embodiments, each received communication is interrogated by a plurality of interrogation engines where each such interrogation engine is of a particular type designed to test the communication for a particular security risk. Each received communication is interrogated by a series of interrogation engines of differing types. The ordering and selection of interrogation engine types for use with received communications may, in some embodiments, be configurable, whereas in others the ordering and selection may be fixed.

Associated with each interrogation engine is a queue of indices for communications to be evaluated by the particular interrogation engine. When a communication is received, it is stored and assigned an index. The index for the receive communication is placed in a queue associated with an interrogation of a particular type as determined by the interrogation engine ordering. Upon completion of the assessment of the received communication by the interrogation engine associated with the assigned queue, the index is assigned to a new queue associated with an interrogation engine of the next type as determined by the interrogation engine ordering. The assignment process continues until the received communication has been assessed by an interrogation engine of each type as determined by the interrogation engine selection. If the communication successfully passes an interrogation engine of each type, the communication is forwarded to its appropriate destination. In some embodiments, if the communication fails any particular engine, a warning indicator signal may be output; in some such embodiments, the communication may then be

forwarded with or without an indication of its failure to its appropriate destination, to an application administrator and/or both.

In some embodiments, the system can use a selected secure delivery mechanism for forwarding of the message to its destination. Such secure delivery can occur as an independent standalone system requiring none of the interrogation and/or threat management features discussed above or below, or may work within an integrated environment providing some, or all, of the discussed features.

The present invention can provide secure communication services by securing a communication link (channel-layer security) or by securing the data sent (message-layer security). Channel-layer and message-layer security can be provided separately or in combination. Some preferred embodiments of the present invention can provide a channel-layer security using SSL or TLS. One preferred embodiment of the present invention can send and/or receive electronic communications including, as a non-limiting example, e-mail and Web requests and responses. In addition, or alternatively, the present invention can send and/or receive other forms of electronic communication such as instant messages, files, pager messages, and voice using applicable protocols.

Accordingly, one preferred method of secure delivery includes a variety of steps that may, in certain embodiments, be executed by the environment summarized above and more fully described below or be stored as computer executable instructions in and/or on any suitable combination of computer-readable media. An electronic communication is received. In some embodiments, a determination is made as to whether the received communication requires secure delivery; in some such embodiments, a configured default may require secure delivery. In other embodiments, no determination need be made as secure delivery is assumed. Secure delivery of the communication to a predetermined recipient is attempted. The attempt uses a secure delivery mechanism selected from a group of one or more available delivery mechanisms. In some embodiments, if the first attempt fails, a second delivery mechanism is chosen and an attempt to deliver via this second mechanism occurs. In some such embodiments, repeated attempts to deliver by different mechanisms may occur until successful delivery or until exhaustion of available mechanisms.

In some embodiments using this queuing approach, the assignment of an index for a received communication to a queue for an interrogation engine of a particular type may involve an evaluation of the current load across all queues for the particular interrogation engine type. If a threshold load exists, a new instance of an interrogation engine of the particular type may be spawned with an associated index queue. The index for the received communication may then be assigned to the queue associated with the interrogation engine instance. In some embodiments, the load across the queues associated with the particular type may be redistributed across the queues including the one associated with the new interrogation engine instance prior to the assignment of the index associated with the newly received communication to the queue. Some embodiments may also periodically, or at particular times such as a determination that a particular queue is empty, evaluate the load across queues for a type of interrogation engine and if an inactivity threshold is met, shutdown excess interrogation instances of that type and disassociating or deallocating indices queues associated with shutdown instances.

Alternatively, a fixed number of interrogation engines of each particular type may be configured in which case dynamic instance creation may or may not occur. In fixed instance embodiments not supporting dynamic instance creation, assignment to a particular queue may result from any appropriate allocation approach including load evaluation or serial cycling through queues associated with each interrogation engine instance of the particular type desired.

In some embodiments, anomaly detection may occur through a process outlined as follows. In such a process, data associated with a received communication is collected. The data may be accumulated from a variety of source such as from the communication itself and from the manner of its transmission and receipt. The data may be collected in any appropriate manner such as the multiple queue interrogation approach summarized above and discussed in greater detail below. Alternatively, the data collection may result from a parallel testing process where a variety of test is individually applied to the received communication in parallel. In other embodiments, a single combined analysis such as via neural network may be applied to

simultaneously collect data associated with the received communication across multiple dimensions.

The collected data is then analyzed to determine whether the received communication represents an anomaly. The analysis will typically be based upon the collected data associated with the received communication in conjunction with established communication patterns over a given time period represented by aggregated data associated with previously received communications. The analysis may further be based upon defined and/or configurable anomaly rules. In some embodiments, analysis may be combined with the data collection; for instance, a neural network could both collect the data associated with the received communication and analyze it.

The adaptive communication interrogation can use established communication patterns over a given time period represented by aggregated data associated with previously received communications. The analysis can further be based upon defined and/or configurable spam rules. In some embodiments, analysis can be combined with the data collection; for instance, a neural network could both collect the data associated with the received communication and analyze it.

Finally, if an anomaly is detected with respect to the received communication, an indicator signal is generated. The generated signal may provide a warning to an application administrator or trigger some other appropriate action. In some embodiments, the indicator signal generated may provide a generalized indication of an anomaly; in other embodiments, the indicator may provide additional data as to a specific anomaly, or anomalies, detected. In the latter embodiments, any warning and/or actions resulting from the signal may be dependent upon the additional data.

Data collected from received communications can be analyzed to determine whether the received communication is on one or more whitelists. The analysis is typically based upon the collected data associated with the received communication in conjunction with reference to one or more whitelists. If no match to a whitelist is found, the communication can be subject to a certain level of interrogation. If a match to the whitelist is found, the communication can either bypass any message interrogation or it can be subject to a different level of interrogation. In one preferred

embodiment, if a match to a whitelist is found, the message can be subject to either adaptive message interrogation or no message interrogation. If no match to a whitelist is found, the message can be subject to normal message interrogation. Additionally, a whitelist can be created and/or updated based on outbound communication. In one preferred embodiment, some or all of the destination addresses of outbound communications are added to a whitelist. If a destination address already appears on a whitelist, a confidence value associated with the destination can be modified based upon the destination address' presence. For instance, a usage count may be maintained; such a usage count can reflect absolute usage of the address or usage of the address over a given period of time.

Additional advantages of the invention will be set forth in part in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description, serve to explain the principles of the invention.

FIG. 1 depicts a typical prior art access environment.

FIG. 2 depicts a hardware diagram for an environment using one preferred embodiment according to the present invention.

FIG. 3 is a logical block diagram of the components in a typical embodiment of the present invention.

FIG. 4 is a flow chart of an exemplary anomaly detection process according to the present invention.

FIG. 5 is a sample anomaly detection configuration interface screen.

FIG. 6 is a bock diagram depicting the architecture of an exemplary embodiment of a security enhancement system according to the present invention.

FIG. 7 is a block diagram depicting the architecture of an exemplary embodiment of a risk assessment approach according to the present invention using multiple queues to manage the application of a plurality of risk assessments to a received communication.

FIGs. 8A-8B are a flow chart depicting the process of accessing risk associated with a received communication using the architecture depicted in FIG. 7.

FIG. 9 is a flow chart of an exemplary communication assessment process according to the present invention.

FIG. 10 is a flow chart of an exemplary whitelist management process according to the present invention.

FIG. 11 is a flow chart of an exemplary interrogation process according to the present invention.

FIG. 12 depicts an overview of information flow through one preferred embodiment of the threat management architecture.

FIG. 13 depicts a block diagram of the Threat Management Center (TMC) using one preferred embodiment according to the present invention.

FIG. 14 depicts an exemplary Threat Pushback System using one preferred embodiment according to the present invention.

FIG. 15 depicts components of a typical individual Messaging Security System (or application layer security system) according to the present invention.

FIG. 16 is a flow chart of an exemplary secure delivery process according to the present invention.

FIG. 17 is a flow chart of a further exemplary secure delivery process according to the present invention.

DETAILED DESCRIPTION

Exemplary embodiments of the present invention are now described in detail. Referring to the drawings, like numbers indicate like parts throughout the views. As used in the description herein and throughout the claims that follow, the meaning of

"a," "an," and "the" includes plural reference unless the context clearly dictates otherwise. Also, as used in the description herein and throughout the claims that follow, the meaning of "in" includes "in" and "on" unless the context clearly dictates otherwise. Finally, as used in the description herein and throughout the claims that follow, the meanings of "and" and "or" include both the conjunctive and disjunctive and may be used interchangeably unless the context clearly dictates otherwise.

Ranges may be expressed herein as from "about" one particular value, and/or to "about" another particular value. When such a range is expressed, another embodiment includes from the one particular value and/or to the other particular value. Similarly, when values are expressed as approximations, by use of the antecedent "about," it will be understood that the particular value forms another embodiment. It will be further understood that the endpoints of each of the ranges are significant both in relation to the other endpoint, and independently of the other endpoint.

Architecture of a Typical Access Environment

FIG. 12 depicts an overview of information flow through one environment using various aspect of the threat management architecture of the present invention. At the Message Security System (MSS) (e.g., 1205), statistics can be collected based on traffic and threat patterns. The statistics can be processed locally by an individual MSS, or they can be processed by an external processor. An MSS is an example of an application layer security system such as hardware device 210. Detailed information can be sent from one or more MSS back to the Threat Management Center (TMC) 1210. In some embodiments, information can be shared among MSSs in the network. In some embodiments, a plurality of MSSs may operate as a peer-to-peer system 1215. In a preferred embodiment, information gathered and/or computed statistics can be sent from a MSS 1205 to the threat notification and pushback system 1220.

Application Layer Security System

FIG. 15 depicts message flow through an exemplary Message Interrogation Engine (MIE) 4035, described in greater detail herein below. The MIE can use rules and policies to perform interrogation. Input from the TMC can be added to the set of rules and policies 4005. The MIE produces a set of statistics 4010 based on its

recorded history. The statistics processing module (SPM) 4025 can process this information and prepare it for distribution. Certain information can be sent back to the TMC for analysis 4030. Information can be sent to and from peers 4015 as part of the peer-based threat notification system. Information is also pushed to towards the source using the threat pushback system 4020. The SPM 4010 can also receive input from the peer-based threat notification system and the threat pushback system. Based on its history and analysis, the SPM 4010 can create new rules and policies 4005 for the local MIE.

FIG. 2 depicts a typical environment according to the present invention. As compared with FIG. 1, the access environment using systems and methods according to the present invention may include a hardware device 210 connected to the local communication network such as Ethernet 180 and logically interposed between the firewall system 140 and the local servers 120 and clients 130. All application related electronic communications attempting to enter or leave the local communications network through the firewall system 140 are routed to the hardware device 210 for application level security assessment and/or anomaly detection. Hardware device 210 need not be physically separate from existing hardware elements managing the local communications network. For instance, the methods and systems according to the present invention could be incorporated into a standard firewall system 140 or router (not shown) with equal facility. In environment not utilizing a firewall system, the hardware device 210 may still provide application level security assessment and/or anomaly detection.

For convenience and exemplary purposes only, the foregoing discussion makes reference to hardware device 210; however, those skilled in the art will understand that the hardware and/or software used to implement the systems and methods according to the present invention may reside in other appropriate network management hardware and software elements. Moreover, hardware device 210 is depicted as a single element. In various embodiments, a multiplicity of actual hardware devices may be used. Multiple devices that provide security enhancement for application servers of a particular type such as e-mail or Web may be used where communications of the

particular type are allocated among the multiple devices by an appropriate allocation strategy such as (1) serial assignment that assigns a communication to each device sequentially or (2) via the use of a hardware and/or software load balancer that assigns a communication to the device based upon current device burden. A single device may provide enhanced security across multiple application server types, or each device may only provide enhanced security for a single application server type.

In one embodiment, hardware device 210 may be a rack-mounted Intel-based server at either 1U or 2U sizes. The hardware device 210 can be configured with redundant components such as power supplies, processors and disk arrays for high availability and scalability. The hardware device 210 may include SSL/TLS accelerators for enhanced performance of encrypted messages.

The hardware device 210 will include a system processor potentially including multiple processing elements where each processing element may be supported via Intel-compatible processor platforms preferably using at least one PENTIUM III or CELERON (Intel Corp., Santa Clara, CA) class processor; alternative processors such as UltraSPARC (Sun Microsystems, Palo Alto, CA) could be used in other embodiments. In some embodiments, security enhancement functionality, as further described below, may be distributed across multiple processing elements. The term processing element may refer to (1) a process running on a particular piece, or across particular pieces, of hardware, (2) a particular piece of hardware, or either (1) or (2) as the context allows.

The hardware device 210 would have an SDS that could include a variety of primary and secondary storage elements. In one preferred embodiment, the SDS would include RAM as part of the primary storage; the amount of RAM might range from 128 MB to 4 GB although these amounts could vary and represent overlapping use such as where security enhancement according to the present invention is integrated into a firewall system. The primary storage may in some embodiments include other forms of memory such as cache memory, registers, non-volatile memory (e.g., FLASH, ROM, EPROM, etc.), etc.

The SDS may also include secondary storage including single, multiple and/or varied servers and storage elements. For example, the SDS may use internal storage devices connected to the system processor. In embodiments where a single processing element supports all of the security enhancement functionality, a local hard disk drive may serve as the secondary storage of the SDS, and a disk operating system executing on such a single processing element may act as a data server receiving and servicing data requests.

It will be understood by those skilled in the art that the different information used in the security enhancement processes and systems according to the present invention may be logically or physically segregated within a single device serving as secondary storage for the SDS; multiple related data stores accessible through a unified management system, which together serve as the SDS; or multiple independent data stores individually accessible through disparate management systems, which may in some embodiments be collectively viewed as the SDS. The various storage elements that comprise the physical architecture of the SDS may be centrally located, or distributed across a variety of diverse locations.

The architecture of the secondary storage of the system data store may vary significantly in different embodiments. In several embodiments, database(s) are used to store and manipulate the data; in some such embodiments, one or more relational database management systems, such as DB2 (IBM, White Plains, NY), SQL Server (Microsoft, Redmond, WA), ACCESS (Microsoft, Redmond, WA), ORACLE 8i (Oracle Corp., Redwood Shores, CA), Ingres (Computer Associates, Islandia, NY), MySQL (MySQL AB, Sweden) or Adaptive Server Enterprise (Sybase Inc., Emeryville, CA), may be used in connection with a variety of storage devices/file servers that may include one or more standard magnetic and/or optical disk drives using any appropriate interface including, without limitation, IDE and SCSI. In some embodiments, a tape library such as Exabyte X80 (Exabyte Corporation, Boulder, CO), a storage attached network (SAN) solution such as available from (EMC, Inc., Hopkinton, MA), a network attached storage (NAS) solution such as a NetApp Filer 740 (Network Appliances, Sunnyvale, CA), or combinations thereof may be used. In

other embodiments, the data store may use database systems with other architectures such as object-oriented, spatial, object-relational or hierarchical or may use other storage implementations such as hash tables or flat files or combinations of such architectures. Such alternative approaches may use data servers other than database management systems such as a hash table look-up server, procedure and/or process and/or a flat file retrieval server, procedure and/or process. Further, the SDS may use a combination of any of such approaches in organizing its secondary storage architecture.

The hardware device 210 would have an appropriate operating system such as WINDOWS/NT, WINDOWS 2000 or WINDOWS/XP Server (Microsoft, Redmond, WA), Solaris (Sun Microsystems, Palo Alto, CA), or LINUX (or other UNIX variant). In one preferred embodiment, the hardware device 210 includes a pre-loaded, pre-configured, and hardened UNIX operating system based upon FreeBSD (FreeBSD, Inc., http://www.freebsd.org). In this embodiment, the UNIX kernel has been vastly reduced, eliminating non-essential user accounts, unneeded network services, and any functionality that is not required for security enhancement processing. The operating system code has been significantly modified to eliminate security vulnerabilities.

Depending upon the hardware/operating system platform, appropriate server software may be included to support the desired access for the purpose of configuration, monitoring and/or reporting. Web server functionality may be provided via an Internet Information Server (Microsoft, Redmond, WA), an Apache HTTP Server (Apache Software Foundation, Forest Hill, MD), an iPlanet Web Server (iPlanet E-Commerce Solutions - A Sun - Netscape Alliance, Mountain View, CA) or other suitable Web server platform. The e-mail services may be supported via an Exchange Server (Microsoft, Redmond, WA), sendmail or other suitable e-mail server. Some embodiments may include one or more automated voice response (AVR) systems that are in addition to, or instead of, the aforementioned access servers. Such an AVR system could support a purely voice/telephone driven interface to the environment with hard copy output delivered electronically to suitable hard copy output device (e.g., printer, facsimile, etc.), and forward as necessary through regular mail, courier, inter-office mail, facsimile or other suitable forwarding approach. In one preferred

embodiment, an Apache server variant provides an interface for remotely configuring the hardware device 210. Configuration, monitoring, and/or reporting can be provided using some form of remote access device or software. In one preferred embodiment, SNMP is used to configure and/or monitor the device. In one preferred embodiment, any suitable remote client device is used to send and retrieve information and commands to/from the hardware device 210. Such a remote client device can be provided in the form of a Java client or a Windows-based client running on any suitable platform such as a conventional workstation or a handheld wireless device or a proprietary client running on an appropriate platform also including a conventional workstation or handheld wireless device.

Application Layer Electronic Communication Security Enhancement

FIG. 3 depicts a block diagram of the logical components of a security enhancement system according to the present invention. The overall analysis, reporting and monitoring functionality is represented by block 310, and anomaly detection is represented by block 370.

Blocks 320-360 represent different assessments that may be applied to electronic communications. These blocks are representative of assessments that may be performed and do not constitute an exhaustive representation of all possible assessments for all possible application server types. The terms "test" and "testing" may be used interchangeably with the terms "assess", "assessment" or "assessing" as appropriate in the description herein and in the claims that follow.

- Application specific firewall 320 provides functionality to protect against application-specific attacks. For instance in the context of e-mail, this assessment could protect against attacks directed towards Extended SMTP, buffer overflow, and denial of service.
- Application specific IDS 330 provides real-time monitoring of activities specific to the application server. This may also retrieve information from multiple layers including the application layer, network layer and operating system layer. This compliments a network intrusion detection system by adding an additional layer of application specific IDS monitoring.

- Application specific anti-virus protection and anti-spam protection 340 provides support for screening application specific communications for associated viruses and/or spam.
- Policy management 350 allows definition of corporate policies with respect to the particular application in regard to how and what application specific communications are sent, copied or blocked. Executable attachments or communication components, often sources of viruses and/or worms, and/or questionable content can be stripped or quarantined before they get to the application server or client. Mail messages from competitors can be blocked or copied. Large messages can be relegated to off-peak hours to avoid network congestion.
- Application encryption 360 provides sending and receiving application communications securely, potentially leveraging hardware acceleration for performance.

The application security system processes incoming communications and appears to network intruders as the actual application servers. This prevents the actual enterprise application server from a direct or indirect attack.

Electronic communications attempting to enter or leave a local communications network can be routed through present invention for assessment. The results of that assessment can determine if that message will be delivered to its intended recipient.

An incoming or outgoing communication, and attachments thereto, are received by a security system according to the present invention. The communication in one preferred embodiment is an e-mail message. In other embodiments, the communication may be an HTTP request or response, a GOPHER request or response, an FTP command or response, telnet or WAIS interactions, or other suitable Internet application communication.

The automated whitelist generation of the present invention allows the system to automatically create and/or maintain one or more whitelists based on the outbound email traffic. In some embodiments, the system can monitor outbound, and/or inbound, email traffic and thereby determine the legitimate email addresses to add to the

whitelist. The software can use a set of metrics to decide which outbound addresses are actually legitimate addresses.

A data collection process occurs that applies one or more assessment strategies to the received communication. The multiple queue interrogation approach summarized above and described in detail below provides the data collection functionality in one preferred embodiment. Alternatively, the assessments may be performed on each received message in parallel. A separate processing element of the system processor would be responsible for applying each assessment to the received message. In other embodiments, multiple risk assessments may be performed on the received communication simultaneously using an approach such as a neural network. The application of each assessment, or the assessments in the aggregate, generates one or more risk profiles associated with the received communication. The risk profile or log file generated based upon the assessment of the received communication is stored in the SDS. The collected data may be used to perform threat analysis or forensics. This processing may take place after the communication is already received and forwarded.

In one preferred embodiment, particular assessments may be configurably enabled or disabled by an application administrator. An appropriate configuration interface system may be provided as discussed above in order to facilitate configuration by the application administrator.

An anomaly detection process analyzes the stored risk profile associated with the received communication in order to determine whether it is anomalous in light of data associated with previously received communications. In one preferred embodiment, the anomaly detection process summarized above and described in detail below supports this detection functionality. Anomaly detection in some embodiments may be performed simultaneously with assessment. For instance, an embodiment using a neural network to perform simultaneous assessment of a received communication for multiple risks may further analyze the received communication for anomalies; in such an embodiment, the data associated with the previously received communications may be encoded as weighting factors in the neural network.

In some embodiments, the thresholds for various types of anomalies may be dynamically determined based upon the data associated with previously received communications. Alternatively, an interface may be provided to an application administrator to allow configuration of particular thresholds with respect to individual anomaly types. In some embodiments, thresholds by default may be dynamically derived unless specifically configured by an application administrator.

Anomalies are typically detected based upon a specific time period. Such a time period could be a particular fixed period (e.g., prior month, prior day, prior year, since security device's last reboot, etc.) and apply to all anomaly types. Alternatively, the time period for all anomaly types, or each anomaly type individually, may be configurable by an application administrator through an appropriate interface. Some embodiments may support a fixed period default for all anomaly types, or each anomaly type individually, which may be overridden by application administrator configuration.

In one preferred embodiment, the stored risk profile associated with the received communication is aggregated with data associated with previously received communications of the same type. This newly aggregate data set is then used in analysis of subsequently received communications of that type.

If an anomaly is detected, an anomaly indicator signal is output. The outputted signal may include data identifying the anomaly detected and the communication in which the anomaly was detected. Various types of anomalies are discussed below with respect to e-mail application security. These types of anomalies may be detected using the specific detection approach discussed below or any of the aforementioned alternative anomaly detection approaches.

The outputted signal may trigger a further response in some embodiments; alternatively, the outputted signal may be the response. In one preferred embodiment, the outputted signal may be a notification to one or more designated recipient via one or more respective, specified delivery platform. For instance, the notification could be in the form of an e-mail message, a page, a facsimile, an SNMP (Simple Network Management Protocol) alert, an SMS (Short Message System) message, a WAP

(Wireless Application Protocol) alert, OPSEC (Operations Security) warning a voice phone call or other suitable message. Alternatively, such a notification could be triggered by the outputted signal.

Using SNMP allows interfacing with network level security using a manager and agent; an example would be monitoring traffic flow through a particular router. OPSEC is a formalized process and method for protecting critical information. WAP is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. An example would be formatting a WAP page to a wireless device that supports WAP when an anomaly is detected. WAP pages are stripped down versions of HTML and are optimized for wireless networks and devices with small displays. SMS is a wireless technology that utilizes SMTP and SNMP for transports to deliver short text messages to wireless devices such as a Nokia 8260 phone. SMS messages could be sent out to these devices to alert a user of an intrusion detection of anomaly alert.

Instead of or in addition to a notification, one or more corrective measures could be triggered by the outputted signal. Such corrective measures could include refusing acceptance of further communications from the source of the received communication, quarantining the communication, stripping the communication so that it can be safely handled by the application server, and/or throttling excessive numbers of incoming connections per second to levels manageable by internal application servers.

In one preferred embodiment, an interface may be provided that allows an application administrator to selectively configure a desired response and associated this configured response with a particular anomaly type such that when an anomaly of that type is detected the configured response occurs.

Finally, if an anomaly is detected with respect to a received communication, the communication may or may not be forwarded to the intended destination. Whether communications determined to be anomalous are forwarded or not may, in certain embodiments, be configurable with respect to all anomaly types. Alternatively, forwarding of anomalous communications could be configurable with respect to individual anomaly types. In some such embodiments, a default forwarding setting

could be available with respect to any individual anomaly types not specifically configured.

Secure Communication Delivery

Communication services performed according to the present invention can be executed on one system processor or they may be distributed across multiple system processors. All relevant functionality of the present invention can be configured by one or more administrators. One preferred embodiment can include an interface for an administrator to perform certificate and user/domain management and feature configuration. A user management interface can be provided for administration of user parameters and certificates. The present invention can be programmed or adapted so that any feature can have an interface for its configuration in a menu-driven system.

In one preferred embodiment with reference to FIG. 16, processing of an inbound communication begins with receipt of an electronic communication designated for delivery to a predetermined recipient 5005. The received communication can, but need not in all embodiments, undergo the testing and analysis described herein above and below regarding threat and/or anomaly detection. A secure delivery mechanism is chosen for use with the received communication 5010. This mechanism is chosen based upon a prioritization of available delivery mechanisms. An attempt is made to deliver the message according to the selected mechanism 5015. If the attempt is successful 5020, the process is complete 5025. In some embodiments, the selection and delivery attempt process can be repeated with an additional selected delivery mechanism upon failure of the attempt to deliver. In some such embodiments, repetition can continue until successful delivery of the communication occurs or all available delivery mechanisms have been attempted. In some embodiments, insecure delivery is performed if no secure delivery mechanism is available or if all available secure delivery mechanisms have failed.

In one preferred embodiment, the delivery mechanism can include a base mechanism and one or more security options. The base mechanism is preferably instant messaging, SMTP, HTTP, or FTP. One skilled in the art will recognize that other commonly used delivery mechanisms can be used as appropriate. Delivery

mechanisms can also include a security option. These options can preferably include SMTP notification with HTTP presentment, S/MIME, PGP, TLS, SSL and combinations thereof; however, other channel and/or message level security technology as known by those skilled in the art could be applied. Delivery mechanisms can use an appropriate message-level encryption technique, a channel-level encryption technique, or a combination of the two. In some embodiments, the available delivery mechanisms can be selected based on the communication source, a predetermined recipient, a default configuration or combinations thereof.

In selecting a secure delivery mechanism, the selection process may include a determination of available secure delivery mechanisms and a prioritization of such mechanisms. The determination and prioritization can in some embodiments occur concurrently with, or as part of, the selection process such as in the case of a default prioritization that serially checks availability of particular delivery mechanisms in the order defined by the default prioritization and chooses the first such available mechanism; FIG. 17 depicts one such process.

In some preferred embodiments, the delivery mechanisms can be prioritized based upon a rating associated with each delivery mechanism. The prioritization of the delivery mechanisms can be retrieved based upon the recipient or the communication source. Additionally, in some embodiments, the communication source can be provided with an interface for designating a prioritization of the plurality of delivery mechanisms. That prioritization information can then be received from the provided interface. In addition, or instead, an interface can be provided to recipients that allow the recipient to specify a default prioritization of delivery mechanism. This configuration information is received from the provided interface and used for secure delivery of communications to that recipient.

In one preferred embodiment, each delivery mechanism can be associated with a rating. The rating can be based upon various criteria including delivery efficiency, delivery cost, delivery security and combinations thereof.

In the alternative, or in addition, an administrator can be provided with an interface for designating a prioritization of the plurality of delivery mechanisms. The

designated prioritization information can then be received from the provided interface. A configuration specified by an administrator can, for example, be used to specify a default prioritization of delivery mechanism.

In some embodiments, a determination is made as to whether the communication requires secure delivery. The requirement of secure delivery can be determined in a number of ways based upon communication source, communication content, communication recipient, configuration data or a combination of such bases. The content of the communication can be parsed for indicia indicating a desire for secure delivery. The received communication can then be specified as requiring secure delivery based upon the parsing. The parsing can be performed by applying one or more filtering rules to the received communication. The filtering rules can be based on content, attachments, sources, recipients, or other indicia one skilled in the art would recognize as appropriate. In some embodiments, the parsing may be limited to a header portion of the communication such as the header portion of an SMTP e-mail message or an HTTP request or response. The received communication can be specified as requiring secure delivery if one or more predetermined keywords are parsed from the received communication, or the header thereof. Instead, or in addition, the source or recipient of the communication can be used to determine secure delivery. For instances, all messaged to a particular recipient or from a particular source could be designated as requiring secure delivery. Further, configuration data submitted by an administrator, or from a communication source, could serve to designate a particular communication for secure delivery.

The present invention can be configured to use or attempt to use any combination of message-level and channel-level security for a communication to be delivered. Recipients and domains can be associated in a database with a variety of secure delivery mechanisms. The delivery mechanisms can be prioritized as described above. A prioritized order can be used based upon the delivery mechanism, the base mechanism of the delivery mechanism, the security option or options of the delivery mechanisms or a combination of these factors. In one preferred embodiment, the prioritized order is based upon security options associated with the delivery

mechanism, from highest preference to lowest: S/MIME, PGP, SWM (Secure Web Mail) with password and random URL, TLS or SSL, SWM with random URL. One skilled in the art will recognize that alternative orders of priority can be used. Such alternative order can include all, some, or none of the secure delivery mechanisms listed above.

In one preferred embodiment, a secure delivery process with a default prioritization based upon security options of PGP, S/MIME, TLS or SSL, SWM (with or without user authentication) is used as depicted in FIG. 17. A communication is received 5100. Selection of a secure delivery mechanism occurs according to an implicit default prioritization of security options as depicted in steps 5105, 5115, 5125, and 5135. A determination of available secure delivery mechanisms occurs concurrently with the selection as follows:

[5105]	Is the recipient capable of receiving a communication using PGP message level encryption?
[5110]	If so, use this mechanism to encrypt the communication and forward the encrypted communication to the recipient in step 5150.
[5115]	If the recipient is not capable of PGP, is the recipient capable of receiving a communication using S/MIME message level encryption?
[5120]	If so, use this mechanism to encrypt the communication and forward the encrypted communication to the recipient in step 5150.
[5125]	If PGP and S/MIME are unavailable, is the recipient capable of receiving a communication using SSL or TLS channel level encryption?
[5130]	If so, use this mechanism to encrypt the communication channel and forward the communication to the recipient in step 5150.
[5135]	If PGP, S/MIME and SSL/TLS do not apply, is the recipient capable of receiving a communication via SWM?
[5140]	If so, copy the communication to the SWM server and generate a notification for delivery to the intended recipient via step 5150. In some embodiments, the notification could be subject to message level and/or channel level encryption.

[5145]	If no secure delivery mechanism is available, deliver the communication via an insecure mechanism.
[5150]	The communication is securely delivered.

In some embodiments, a loop back can occur upon failure of delivery in step 5150. The loop back would lead back to the next available mechanism after the one that failed. For instance, if PGP was determined to be available but delivery failed, the loop back would re-enter the process at step 5115 to determine availability of S/MIME. In such alternatives, the decision to proceed to the next step is based upon success or failure of delivery instead of, or in addition to, availability of the particular security option. In addition, or in alternative embodiments, the insecure delivery step can be totally removed or substituted with a loop back to step 5105 and continue checking for available secure delivery mechanisms until one is found; in addition, a constraint could be place on such a loop back so that the loop only occurred a fixed number of times or occurred for a fixed time period.

The present invention can operate independently of, or in concert with, other networking administration methods and systems as discussed herein above and below. The following components of the present invention can be incorporated into the present invention. Once incorporated, they can be configured separately or in combination.

Secure Web Mail (SWM)

SWM service can send a notification message to a recipient with a link to a secure web page to access an original message. When the provided link is followed, the recipient's browser can establish a secure connection with a server containing the original message. The message can then be viewed securely. The user access to the message through the browser connection can also be secured by appropriate authentication means. As non-limiting examples, the authentication for the user access may be based on a random URL, username/password combination, SecureID, or combinations thereof.

In one preferred embodiment, an administrative interface for the recipient can be provided to configure the maintenance of read and unread messages. As a non-limiting example, the system can be configured to delete message once it is viewed.

For each recipient of a message to be delivered by SWM, a random string can be generated and a URL created using it. For example, the URL https://boxname/msg.jsp?ran=12fugrl980dh89 could be used. The present invention can generate an e-mail with the URL enclosed in it. Such an e-mail can be used to inform a recipient that a message can be retrieved at the included link. Some embodiments can provide additional authentication by using a username/password or secureID authentication mechanism. Some embodiments of the present invention can be configured to perform user authentication before displaying the message containing the browser link.

The SWM can execute on or as one or more processing elements. These processing elements can be part of the system processor of the present invention, or they can be separately provided with a communication link between them and the system processor. In either case the received communication is copied to the SDS of the present invention, or some storage associated with the separately provided processing elements. The location of the copy of the communication can be used in some embodiments to generate the URL; however, other embodiments can generate the URL independently of the copy location.

The SWM provides for secure presentment of the received communication through appropriate encryption using message level encryption (*e.g.*, S-HTTP, S/MIME, PGP, etc.) and/or channel level encryption (*e.g.*, SSL or TLS). In one preferred embodiment, SSL channel level encryption serves as the security option for providing secure presentment. The various public and private keys for use in encryption can be available locally within the SDS of the present invention; in addition, or instead, the system processor can obtain such keys from an appropriate (preferably trusted and authenticated) key server.

Server-based PGP

Embodiments including server-based PGP can be programmed or adapted to send and/or receive server-based PGP messages. Server-based PGP can be used in environments where it is desired to provide confidentiality, authentication and assurances of message integrity. This functionality can be implemented using a PGP

toolkit such as PGP Freeware (Massachusetts Institute of Technology; Cambridge, MA) or provided by Network Associates Technology, Inc. (Santa Clara, CA).

One preferred embodiment can include a configuration interface to administer the domains and/or users that can send and/or receive messages using the PGP/MIME format. The administration interface can also provide facility for public-key management for domains. The present invention can be configured to generate a PGP/MIME message from an original message for those recipients or the domains that are specified to require PGP/MIME message-level encryption. The various public and private keys for use in encryption can be available locally within the SDS of the present invention; in addition, or instead, the system processor can obtain such keys from an appropriate (preferably trusted and authenticated) key server.

Incoming Messages

In one preferred embodiment of the present invention, if the recipient or domain of an incoming message can be matched in the database to a recipient or domain on a PGP user list, then the message can be decrypted using the corresponding public key. Authentication of the sender can also performed.

Outgoing Messages

In a preferred embodiment, the recipients and domains of an outgoing message can also be checked against a PGP user list. If a recipient or destination domain matches a recipient of domain appearing in the PGP user list, then a new communication can be created using the PGP/MIME format. The new communication can be created using the original filename with a .pgp extension, or other suitable naming convention. After creation, a process compliant with the appropriate protocol can deliver the encrypted communication.

Server-based S/MIME (Secure/Multipurpose Internet Mail Extensions)

Some preferred embodiments of the present invention can include the ability to send and receive server-based S/MIME messages. S/MIME can be used to bolster privacy, integrity and provide authentication. The present invention can generate an S/MIME message from an original message recipients and/or domains that are

specified to require the S/MIME format. This feature can be configurably turned on or off.

Server-based S/MIME functionality can be implemented using an S/MIME toolkit. Some preferred embodiments can include a configuration interface to administer domains and users that can receive and/or send messages using the SMIME format. An interface can also be provided for public-key management of the domains. The various public and private keys for use in encryption can be available locally within the SDS of the present invention; in addition, or instead, the system processor can obtain such keys from an appropriate (preferably trusted and authenticated) key server.

Incoming Messages

An incoming message can be checked to see if it is in S/MIME format. If the recipients or the destined domain of a message matches a member of the S/MIME user list, then the message is decrypted using the corresponding public key. Authentication of the sender can also be performed.

Outgoing Messages

A recipient of an outgoing message can be checked against the S/MIME user list. If the recipient or the destination domain requires or supports S/MIME encryption, then a new message can be created based on the S/MIME format. The new S/MIME message can be created using the original filename with a .smime extension, or other suitable naming convention. The present invention can then deliver the S/MIME message to its intended recipients.

SSL/TLS

In some preferred embodiments, messages can be delivered using a secure communication channel employing SSL/TLS or other appropriate secure channel protocol. A recipient of an outgoing message can be checked against the SSL/TLS user list. If the recipient or the destination domain requires or supports channel-level encryption, then the message can be delivered using such a channel. The various public and private keys for use in encryption can be available locally within the SDS of the

present invention; in addition, or instead, the system processor can obtain such keys from an appropriate (preferably trusted and authenticated) key server.

Threat Management Center

A TMC system can reside on a computer system in communication with one or more application and/or network layer security systems. A typical hardware configuration for the TMC includes a system processor and a system data store, which can be similar in capacity to those described herein above with respect to the application layer security systems. Typically, the communication can occur via a computer network such as the Internet; however, one or more systems can connect to the TMC via other mechanism including direct connection and dial-up access.

The TMC includes at least one input, a processing system, and at least one output. FIG. 13 depicts a flow chart of a rule creation process in an exemplary TMC. Input 2005 can be information about messages, messaging systems, attacks, vulnerabilities, threats associated with them, or any other information one skilled in the art would find relevant to threat analysis. In addition, feedback to the TMC may also be provided by one or more application layer security systems. The final output of the TMC 2010 can includes rules and/or policies that can be used to protect against threats, both known and unknown by application layer security systems. These rules and/or policies 2010 can be used by one ore more application layer security systems.

In one preferred embodiment, rule and policy creation can be based on the set of threat information that is received. Information can be received from one or more MSSs or any other threat information source configured to communicate with a TMC.

The output 2010 of the TMC can be influenced by a Rules and Policy Application Programming Interface (API) 2015. While only one API is depicted in the exemplary embodiment of FIG. 13, one skilled in the art will realize that multiple APIs can be configured to perform the functions desired. In some embodiments, the API can be modified as often as necessary or desired to account for any changes in threat and/or traffic patterns. The API can be programmed or adapted to use proprietary formats based on message interrogation systems in place on application layer security servers as well as standard intrusion detection rule formats.

In some embodiments, the output of the Rules and Policy API can be in a natural language. In other embodiments the output can be in a rules expression language including but not limited to regular expressions, intrusion detection information format such as IDMEF, mail filtering languages such as SIEVE, proprietary rule expression formats or other formats one skilled in the art would find appropriate. As a non-limiting example, natural language output can be used to explain to an administrator or user how to configure the system with the suggested rules and policies.

The API can be used to improve the final set of rules generated by the TMC. As a non-limiting example, some message security systems include interrogation engines that use proprietary rule formats. In such a system, a rule to block incoming messages with a "threat.exe" attachment can be specified as:

Attachment Filtering Rule:

Direction: Incoming

Attachment: threat.exe

Action: Drop message

As a non-limiting example, a rule to block incoming messages with a "Threat Title" subject in such a system can be specified as:

Mail Monitoring Rule:

Direction: Both

Field: Subject

Data: Threat Title

Action: Drop message

Different embodiments can use different types of rules for performing different types of filtering. If a Rules and Policy API 2015 is used, the Rule and Policy Creation module 2020 must be programmed or adapted to communicate with the API.

The output 2010 of the TMC can be influenced by goals 2025. The goals 2025 can be global goals, goals for individual messaging security servers, or goals for individual users. As non-limiting examples, some MSS embodiments can have more conservative threat management policies. Some embodiments can be configured to use

rules that are automatically put in place while other embodiments can be configured to use rules to be approved by a local administrator. In some situations it may be desirable to use rules that discard objectionable content while in other situations it may be desirable to quarantine that content. In other situations, a higher or lower confidence in the likelihood of a threat before an action is taken may be desirable. The goals 2025 can be global goals or different goals for different MSSs. As a non-limiting example, the goal may be a certain effectiveness value and a certain accuracy value. For example, a goal can be given to the system that specifies 95% effectiveness and 99.9% accuracy for spam detection.

In some embodiments, as another goal, the system can allow one or more users, MSS, or other entity to provide a definition of threatening communications. As a non-limiting example, in the case of spam, spam may not be well defined. Rather than allowing only a binary decision, the present invention can classify messages in different categories (e.g. business email, personal email, chain letters, adult language, porn, web product offerings, newsletters, mailing lists, etc.) In some embodiments, an individual user, administrator or other suitable human or computerized user can register preferences concerning receipt of any of these types of content. The system can then enforce that policy for that entity. This can be useful in the threat pushback system further described below and depicted in FIG. 14.

Inputs to Rule and Policy creation 2020 can include, but are not limited to the following:

1. Spam and non-spam messages from archives such as SpamArchive.org, user reported spam, spam identified by the individual messaging security systems, information about misclassified messages, information from databases of known spam such as Distributed Checksum Clearinghouse (http://www.rhyolite.com/anti-spam/dcc) and Razor (http://razor.sourceforge.net)
2. Virus information from virus signatures, or other sources of virus information such as virus alert newsletters, and/or virus alert databases. The system can use this information to develop virus information before signatures are available. This information can be obtained from anti-virus vendors Sophos and McAfee, for

example. This information can be retrieved via HTTP, FTP, SMTP, by direct database access, or other appropriate means. In some embodiments, the system can create rules to block virus outbreaks before virus signatures are available as well as for deployments that do not have other anti-virus systems deployed.

3. Intrusion information: This information can be extracted from vulnerability alerts from sources such as bugtraq, CERT, software vendors, open-source projects, information sharing projects such as the FBI InfraGard, or other sources as appropriate. The information can also be received from distributed intrusion detection systems or it can be manually entered by users.

The system can perform input parsing and feature extraction according to input type and source. In the case of spam messages, the input can include spam messages that are stored in proprietary formats such as Microsoft's .pst format, Unix mbox format, forwarded spam or spam sent as an attachment, an archive of spam messages, or other source. The spam messages can be accessed from local storage or from remote storage using protocols such as, but not limited to, HTTP, SCP, FTP, POP, IMAP, or RPC. For an individual message, relevant features can include headers, origin, and message contents. Each type of feature can be extracted and stored as appropriate.

One preferred embodiment can use regular expression content matching tools to parse messages and extract features. A prefilter can be used that defines the regular expression used for content matching. This determines the type of features that are extracted. As a non-limiting example, for extracting message subjects, a regex filter can be used that only examines subject lines. To extract information about all headers, a regex filter can be used that only observes message headers. Similarly different pre-filters can be used to extract different types of content from the body. A normal tokenizer pre-filter can provide normal content features. These features can be words, phrases, n-grams, or other features one skilled in the art would find useful. The prefilter can be sensitive to certain types of words including ignoring certain email address and domain names. The prefilter can also cause the features to focus on email addresses, URLs, phone numbers, etc.

The system can include an anonymization module that assures that sensitive features are not extracted and exposed. As a non-limiting example, the anonymization module can determine the identification of the spam victim and the domain and prevent exposure of that information.

For virus alerts, input can be email messages that explain the presence and properties of a new virus or worm. The input parser can be configured to parse these messages and determine the relevant properties of a threat. These properties can include, but are not limited to, the attachment name or types, subject lines, and from addresses. The input parser may be given different format definition files or pre-filters for the different sources of virus alerts. Alternatively, the virus alert parser can be in communication with web pages to access other information. The information can be parsed for relevant properties. In other embodiments, the virus alert parser may interact directly with a database that stores such information or a user may manually enter data based on such information into the system.

The rule creation system 2020 can reside on a single system or multiple processes can run on multiple systems. Threat information can be reduced to a canonical form and the relevant features extracted. The system can utilize a diversity of algorithms to determine the relevant features and/or reduce the feature set. In some embodiments, each located feature can be associated with an interrogation system on the MSS. The TMC can determine the appropriate type of rule to create. In some embodiments, a feature can be expressed using a plurality of interrogation systems. In some embodiments, feature sets can be reduced and efficient types of rules can created.

In some embodiments, resultant rules can have a given weight and certain interrogation systems may have some weight in the overall threat value for a particular message. These values can be determined based on the input from the system. Therefore, these values can be adjusted when desired based on new threats, feedback from the MSSs, and other appropriate sources. The MSSs can be programmed or adapted to determine an aggregate threat likelihood based on automatically adjusted weights, or confidence values for each rule and interrogation system, or other relevant

information. In one preferred embodiment, the rule creation system 2020 can include a scheduler that looks for new threat information.

The system first creates a set of candidate rule sets 3030. Before these are distributed, the system can use goal-based testing 2035 to determine the validity of these rules.

Some embodiments of the present invention can test the rules and policies. The test data 2045 may include threatening and non-threatening data. The system can use the test data sets to discover false positives and negatives of the system as well as general system performance. The goals 2040 used for rule creation can also used as input to the testing. Additional goals, including but not limited to, performance goals can be specified for testing. If specified goals are not met, the system can automatically adjust the feature sets, the weights of individual features, the weights of each interrogation system, and any other relevant parameters to reach the goals. Once the correct tuning is achieved, the rule sets can be approved and distributed the MSSs.

Threat Pushback System

Many systems known in the art only address symptoms of an attack in the local environment. Besides notifying other systems that participate in the network of MSSs, some embodiments of the present invention can determine the source of a threat and push the threat back towards the source. Once the source of a threat is determined, the system can send messages up the network to other systems in the hierarchy.

A threat pushback system can reside on a computer system as part of, or as a compliment to, an application client, an application layer security system or a TMC. A typical hardware configuration for the threat pushback system includes a system processor and a system data store, which can be similar in capacity to those described herein above with respect to the application layer security systems. Typically, the communication can occur via a computer network such as the Internet; however, one or more other mechanism can be used including direct connection and dial-up access.

FIG. 14 depicts an exemplary threat pushback system. Two threat aware modules 3030, 3035 are depicted. Once a threat is detected locally 3005, the threat information can be passed to a Threat Notification Module (TNM) 3020. The TNM

can pass a threat notification 3040 including information about the threat to the Threat Detection Module of another system 3025. In a preferred embodiment, the TNM of a MSS can pass information to another MSS. In still another embodiment, any TMN at any location in the network such as within, or connected to, client applications/systems, application layer security systems and/or threat management centers can pass information to its parent in the network hierarchy.

The threat notification protocol can be standardized across the participating systems. Some embodiments can include a threat response module 3015 programmed or adapted to respond to the threat notification information.

In a preferred embodiment, the system of the present invention can be programmed or adapted to function at the application-layer. Such an embodiment can be readily deployable. If an underlying network-layer pushback system is operational, the system can utilize some of its functionality to determine the path to a threat. Additionally, the threat notification system can determine the source of the attack. As a non-limiting example, in the case of spam, to determine the source of the attack, message headers must be examined. The system can determine how many of these headers can be trusted. Forged headers can be identified and ignored. This process may include lookups to external databases such as registries of IP and ASN numbers such as ARIN or databases of spam sources such as spamhaus.

Because an attacker may be able to forge the path information that is shown in the communication, the system can process the available information to determine the correct path. This can be accomplished with any combination of application level information, network information, or information from external systems such as IP traceback systems, and other resources known to one skilled in the art. At the application level, an attacker may be able to forge some identifying information. The path determination module can provide the path information to the notification sender module. The path determination module can include a path extraction submodule and a path verification submodule.

In one embodiment, the path extraction submodule can parse the identifying information and provide that as the path information. That information, however, has

not verified and could be inaccurate. In another preferred embodiment, the path verification module can process the extracted path information to determine the valid path information. As a non-limiting example, the path extraction submodule can read message information such as the headers. The Return-Path or Received headers can provide information regarding the path of email servers that a message traveled. The "FROM" header can be used to identify the email address of the sender. The "MAIL FROM" RFC 821 header can be used to indicate the email address of the sender. The "EHLO" RFC 821 header can be used to indicate the domain or hostname of the sender. Other headers and message features may be used including the Message-Id and the actual contents of the message. Call for action information is contact information provided for the receiver such as a reply email address, a URL, a postal address, or a phone number. This information in a message can be used. Other information known to one skilled in the art, including but not limited to the IP address of the network connection, can also be used.

Several verification methods can be used to determine information authenticity. As a non-limiting example, most of the above-mentioned headers are easily forged, so a more reliable source is the Return-Path or Received headers. The goal of the present invention is to determine the longest possible authentic path. In one embodiment of the present invention, only the last header is used since this header represents the actual server that contacted the victim's server. Each Received header contains Received from and Received by information. These fields can be verified with DNS for appropriate MX records, A records, and/or reverse records, as well as other appropriate sources known to one skilled in the art. These hosts can be checked against open relay lists, dial-up addresses lists and known spam sources lists. The presence on any of these lists can provide additional information about the last accurate Received header. Additionally, the chain of received headers can be verified against each other. Inconsistencies in this chain can also give additional information about the last accurate Received header. Other details of these headers can be used to verify the path. As a non-limiting example, the date information and server version information can be used.

Once the system determines that a pushback message needs to be forwarded in a particular direction, the system can determine what information needs to be included in the pushback message. The threat notification of the present invention includes additional detailed information about the threat in addition to the IP address of the source.

Detailed threat information can allow systems to make local decisions about how to react to a threat. As a non-limiting example, the above described threat classification system can be used to process spam messages. Information concerning a spam attack sent through the threat pushback system can include information concerning the category of threat and other relevant characteristics.

The receiving system can be configured to block certain portions of communications at an organizational level. Furthermore, ISPs could use this information to block certain categories of spam messages including, but not limited to, fraudulent messages. The system can be configured such that an organization can have policies to block chain letters and adult language. At the desktop level, an individual can configure the system to block newsletters and mailing lists in addition. This allows a common definition for blocked material as close to the source as possible while not requiring a common definition of spam.

The threat information can indicate, among other parameters, the presence of a threat, as well as identify the source, and/or provide detailed threat and/or response information. To identify the source, the information provided can include the identity of the source such as its IP address or hostname, path information, entire determined path information. Additionally, path information can be provided so that other hosts can perform independent own extraction and/or verification. The system of the present invention can also indicate the traffic that is determined to be a threat so that the receivers on the path can determine the details from stored information. This systems and methods of the present invention are an enhanced form of reverse path forwarding used in routing systems.

Whitelisting

In one embodiment, the system can be configured so that communications matched to a whitelist entry may be subject to either no interrogation or less rigorous interrogation. Once a whitelist has at least one entry, the incoming message interrogation system can utilize it in connection with the interrogation of a message.

FIG. 10 depicts operations that can be performed on a whitelist to add an entry. Once an outgoing address passes any exclusion conditions 1005 described above, it can be added to a whitelist. The whitelist can be stored on the SDS. The system first checks to see if the address is already present on the list 1010. If present, the list can be updated with any new information 1015. Before new information is updated, the system can check for sufficient space in the SDS 1025. If sufficient space is not available, additional space is allocated from the SDS 1030. If an address is not found in a whitelist, an initial record can be added for that address. Before a new address is added to a whitelist 1040, the system can check for sufficient space in the SDS 1020. If sufficient space is not available, additional space is allocated from the SDS 1035. In many embodiments, explicit space allocation need not occur rather implicit space allocation occurs as a result of an information update 1015 or an add entry 1040.

The initial record for an outbound address can include the email address, the internal email address, the message sent time, usage count, last time used and/or any other characteristics one skilled in the art would find relevant or useful. In the case of an email address that is already present on a whitelist, the system can use a separate record for each instance of that email address being used as an outbound address or the system can maintain a single record for each outbound address with a summary of information in that entry, including information describing instances of use. The system can store records in a number of other ways using different data structures. The records may include other representations of data in addition to the email address, including by not limited to a hash of the email address.

In a preferred embodiment, the system can store records in a MySQL database. As a non-limiting example, the following command can be used to build a database

comprising the external and internal email addresses, date of last update, and an occurrence counter.

```
create table ct_whitelist
(out_emailaddress    varchar(255) not null,    - External email address
in_emailaddress      varchar(255) not null,    - Internal email address
lastupdatetime       datetime,                 - Last update of this address
curr_count           integer,                  - Address occurrence counter
);
```

Maintaining the Whitelist

In some embodiments, the system can allow unlimited storage. In other embodiments, the storage available for the list can be limited. In still other embodiments, the system can allow for management of the size of the list. A number of caching techniques can be used, including but not limited to first in first out and least recently used. Other techniques can include an accounting of the number of internal users that reported the outbound address. List cleanup can occur in real-time or periodically. Additionally, one skilled in the art will recognize that a wide variety of list management techniques and procedures can be used to manage a whitelist in connection with the present invention.

Whitelist Usage

An example of a system using a whitelist according to the present invention is shown in FIG. 9. One or more relevant parameters of inbound communication 905 are compared against one or more whitelists 910. In some embodiments, the whitelist is checked at each incoming email message. In a preferred embodiment, the comparison includes origination email addresses. If the check against a whitelist 910 reveals no match, then the message is subject to normal message interrogation 915. Normal message interrogation can employ analysis criteria that are the most sensitive to spam or other threats as discussed hereinabove. If a message passes normal interrogation 915, i.e. it is determined not to be spam or a threat (or to have a lower likelihood of being spam or a threat), it can be presented to its intended recipient for delivery 920. If the check against a whitelist 910 reveals a match, the system can be configured to

process the message in a variety of ways. In one embodiment, the system can be programmed or arranged to bypass 925 any message interrogation and deliver the message to its intended recipient 920. In an alternative embodiment, the system can be programmed or arranged to process the message using adaptive message interrogation 930. If adaptive message interrogation 930 determines a message is not spam, it can forward the message for delivery 920.

In some embodiments, both options 925, 930 are selectively available. The decision whether to pass whitelisted communications through adaptive message interrogation 930 or to bypass any message interrogation 925 can be made per deployment or can be based on the details of the whitelist entry. For instance, messages from more frequently used outbound address can bypass 925 interrogation completely whereas messages from less frequently used outbound addresses can be subjected to adaptive message interrogation 930.

If the message goes through normal or adaptive interrogation with the whitelist information, the interrogation module can utilize the whitelist information to effect the type and/or level of interrogation. In some preferred embodiments, the adaptive message interrogation can use multiple levels of trust, as further described below and in FIG. 11. In other embodiments, the adaptive message interrogation can set a confidence indicator indicative of the confidence the interrogator has in its characterization.

Messages that are not delivered to the intended recipient can be either quarantined or deleted. In an alternative embodiment, messages determined to be spam can be indicated as spam or a threat and forwarded to the intended recipient.

Additionally, each outbound email address can be assigned a confidence value. According to the confidence value associated with a given incoming email address, incoming messages can be subjected to variable levels of interrogation. In one preferred embodiment, incoming messages associated with lower confidence values are subjected to more aggressive spam interrogation and incoming messages associated with higher confidence values are subjected to less aggressive spam interrogation. In

other embodiments, the message can be given positive credits to offset any negative spam detection points based on the confidence value.

One preferred embodiment of the system allows some or all external email recipients to be whitelisted 935. Some embodiments can have a metric that describes the number of outgoing messages to a particular email address. When the metric reaches a certain threshold, the email address can be whitelisted. Other embodiments can include the ability to track addresses over time. In those embodiments, if the metric exceeds a certain value for a particular outbound email address during a particular time, then that entry can be whitelisted.

The parameters described above may be configurable by an application administrator through an appropriate interface. Some embodiments may support fixed parameters which may be overridden by application administrator configuration.

In some embodiments, the threshold for characterization as spam or a threat may be dynamically determined based upon the data associated with previously received communications. Alternatively, an interface may be provided to an application administrator to allow configuration of particular thresholds with respect to individual addresses. In some embodiments, thresholds by default may be dynamically derived unless specifically configured by an application administrator.

When spam or a threat is detected, instead of, or in addition to, a notification, one or more response measures could be triggered. Such responsive measures could include refusing acceptance of further communications from the source of the received communication, quarantining the communication, stripping the communication so that it can be forwarded to its intended recipient, and/or throttling excessive numbers of incoming communications from certain sources.

Authenticated Whitelist

One issue with whitelists is that attackers or spammers can pretend to send messages from whitelisted addresses and therefore bypass filtering and anti-spam tools. It is relatively easy for an attacker to forge the sender information on messages. To overcome this limitation of whitelists, the system of the present invention allows the authentication of the sender information. There are several methods for integrating

sender authentication with a whitelist system. In one embodiment, only authenticated senders can be whitelisted. Such a procedure can reduce the likelihood of forged senders being whitelisted. However, in many environments, the percentage of messages that are authenticated is low, thereby reducing the effectiveness of whitelisting. Some embodiments of the present invention can allow both authenticated and unauthenticated senders to be whitelisted. In these embodiments, a higher trust value is given to messages from authenticated senders. SMIME and PGP offer mechanism for providing authentication.

One such embodiment is depicted in FIG. 11. As a non-limiting example, when a message 1105 is received from a sender on a whitelist 1115 an associated level of trust is retrieved or calculated 1135. In some embodiments, the trust level value is a single value associated with the whitelist entry that simply requires retrieval. In other embodiments, the trust level value can be calculated as a weighted sum of various characteristics of the entry; in some such embodiments, the weights can be statically defined, defaulted subject to override by a user or other computer system or dynamically configurable. That associated level of trust can be compared to a threshold level 1140. Any communications that have a trust level that meets or exceeds the trust level threshold can bypass message interrogation 1120 while communications that do not have a trust sufficient trust level will be processed with at least some interrogation 1125. Messages that bypass interrogation 1120 as well as messages that pass interrogation 1125 can be delivered to the intended recipient 1145. In such an embodiment, messages not associated with a whitelist entry are subjected to interrogation and further processing 1150.

Some embodiments of the present invention can allow the trust level threshold 1130 to be configured by an administrator, other user of the system or other computer systems.

Exclusions from Whitelist

The spam/threat detection according to present invention examines every outbound message and maintains a list of known outbound email addresses. The resulting list can then be used as the list of trusted senders. However, it may not be

advisable in all cases to add every outbound message recipient to the list of trusted senders for incoming mail. For example, while a user may send a message to a newsgroup, that does not indicate that messages from this newsgroup should necessarily bypass mail filtering. To further illustrate, a user may send an unsubscribe message to a newsletter or in response to a spam message. Thus, there can be situations in which unconditional whitelist addition is not advisable. The system of the present invention allows certain exclusion conditions to be entered and applied.

These exclusion conditions can include rule sets, heuristics, artificial intelligence, decision trees, or any combination thereof. The conditions can be set by and administrator or other user of the system.

Multiple Queue Approach to Interrogation of Electronic Communications

With reference to FIG. 7, a multiple queue approach is provided for applying a plurality of risk assessments to a received communication.

Messages are first placed in an unprocessed message store 730, a portion of the SDS, for advanced processing and administration. Messages come in from an external source 740 and are placed in this store 730. This store 730 maintains physical control over the message until the end of the process or if a message does not pass interrogation criteria and is, therefore, quarantined.

An index to the message in the store 730 is used to pass through each of the queues 771B, 781B-784B, 791B in the queuing layer 720 and to the interrogation engines 771A, 781A-784A, 791A instead of the actual message itself to provide scalability and performance enhancements as the index is significantly smaller than the message itself.

Both the queues and the interrogation engines use the index to point back to the actual message in the unprocessed message store 730 to perform actions on the message. Any suitable index allocation approach may be used to assign an index to a received message, or communication. For instances, indices may be assigned by incrementing the index assigned to the previously received communication beginning with some fixed index such as 0 for the first received communication; the index could be reset to the fixed starting point after a sufficiently large index has been assigned. In

some embodiments, an index may be assigned based upon characteristics of the received communication such as type of communication, time of arrival, etc.

This approach provides independent processing of messages by utilizing a multi-threaded, multi-process methodology, thereby providing a scalable mechanism to process high volumes of messages by utilizing a multi-threaded, multi-process approach.

By processing messages independently, the queuing layer 720 decides the most efficient means of processing by either placing an index to the message on an existing queue or creating a new queue and placing the index to the message on that queue. In the event that a new queue is created, a new instance of the particular interrogation engine type will be created that will be acting on the new queue.

Queues can be added or dropped dynamically for scalability and administration. The application administrator can, in one preferred embodiment, configure the original number of queues to be used by the system at start-up. The administrator also has the capability of dynamically dropping or adding specific queues or types of queues for performance and administration purposes. Each queue is tied to a particular interrogation engine where multiple queues and multiple processes can exist.

Proprietary application-specific engines can act on each queue for performing content filtering, rules-based policy enforcement, and misuse prevention, etc. A loosely coupled system allows for proprietary application-specific applications to be added enhancing functionality.

This design provides the adaptive method for message interrogation. Application-specific engines act on the message via the index to the message in the unprocessed message store for completing content interrogation.

Administration of the queues provides for retrieving message details via an appropriate interface such as a Web, e-mail and/or telephone based interface system as discussed above in order to facilitate access and management by the application administrator. Administration of the queues allows the administrator to select message queue order (other than the system default) to customize the behavior of the system to best meet the needs of the administrator's particular network and system configuration.

FIGs. 8A - 8B are flow charts depicting use of the multiple queue approach to assess risk associated with a received communication. At step 802 a determination is made if the start-up of the process is being initiated; if so, steps 805 and 807 are performed to read appropriate configuration files from the SDS to determine the type, number and ordering of interrogation engines and the appropriate queues and instances are created. If not, the process waits at step 810 for receipt of a communication.

Upon receipt at step 812, the communication is stored in a portion of the SDS referred to as the unprocessed message store. The communication is assigned at step 815 an index used to uniquely identify it in the unprocessed message store, and this index is placed in the first queue based upon the ordering constraints.

The processing that occurs at step 810 awaiting receipt of communication continues independently of the further steps in this process, and will consequently spawn a new traversal of the remainder of the flow chart with each received communication. In some embodiments, multiple instances of step 810 may be simultaneously awaiting receipt of communications.

In some embodiments, the receipt of a communication may trigger a load evaluation to determine if additional interrogation engines and associated queues should be initiated. In other embodiments, a separate process may perform this load analysis on a periodic basis and/or at the direction of an application administrator.

The index moves through the queue 820 until it is ready to be interrogated by the interrogation engine associated with the queue as determined in step 825. This incremental movement is depicted as looping between steps 820 and 825 until ready for interrogation. If the communication is not ready for evaluation at step 825, the communication continues moves to move through the queue at step 820. If the communication is ready, the index is provided to the appropriate interrogation engine at step 830 in FIG. 8B.

The interrogation engine processes the communication based upon its index in step 830. Upon completion of interrogation in step 835, the interrogation creates a new risk profile associated with the received communication based upon the interrogation.

If additional interrogations are to occur (step 840), the index for the communication is place in a queue for an instance of the next interrogation type in step 845. Processing continues with step 820 as the index moves through this next queue.

If no more interrogations are required (step 840), a further check is made to determine if the communication passed interrogation by all appropriate engines at step 850. If the communication passed all interrogations, then it is forwarded to its destination in step 855 and processing with respect to this communication ends at step 870.

If the communication failed one or more interrogation as determined at step 850, failure processing occurs at step 860. Upon completion of appropriate failure processing, processing with respect to this communication ends at step 870.

Failure processing may involve a variety of notification and/or corrective measures. Such notifications and/or corrective measures may include those as discussed above and in further detail below with respect to anomaly detection.

Anomaly Detection Process

The Anomaly Detection process according to an exemplary embodiment of the present invention uses three components as depicted in FIG. 6:

1. Collection Engine

This is where the actual collection of data occurs. The collection engine receives a communication directed to or originating from an application server. One or more tests are applied to the received communication. These one or more tests may correspond to the various risk assessments discussed above.

The collection engine in one preferred embodiment as depicted in FIG. 6 uses the multiple queue approach discussed above; however, this particular collection engine architecture is intended as exemplary rather than restrictive with respect to collection engines usable within the context of this anomaly detection process.

As depicted in FIG. 6, the collection engine includes one or more interrogation engines of one or more interrogation engine types in an interrogation layer 610. Associated with each interrogation engine type in a queuing layer 620 is at least one indices queue containing the indices of received communication awaiting interrogation

by an interrogation engine of the associated type. Collectively, the queuing layer 620 and the interrogation layer 610 form the collection engine. A received communication is received, stored in the SDS and assigned an index. The index is queued in the queuing layer for processing through the collection engine.

2. Analysis Engine

The data collected by the previous component is analyzed for unusual activity by the anomaly detection engine 640. The analysis is based on data accumulated from analysis of previously received communications over a period of time. A set of predefined heuristics may be used to detect anomalies using dynamically derived or predetermined thresholds. A variety of anomaly types may be defined generally for all types of Internet application communications while others may be defined for only particular application types such as e-mail or Web. The data associated with previously received communications and appropriate configuration data 630 are stored in the SDS.

The set of anomaly types that the analysis engine will detect may be selected from a larger set of known anomaly types. The set of interest may be set at compile time or configurable at run time, or during execution in certain embodiments. In embodiments using the set approach all anomaly types and configuration information are set within the analysis engine. In some such embodiments, different sets of anomalies may be of interest depending upon the type of communication received. In configurable at run time embodiments, anomaly types are read from a configuration file or interactively configured at run time of the analysis engine. As with the set approach, certain anomaly types may be of interest with respect to only selected types of communication. Finally, in some embodiments (including some set or configurable ones), an interface such as described above may be provided allowing reconfiguration of the anomaly types of interest and parameters associated therewith while the analysis engine is executing.

The thresholds for various types of anomalies may be dynamically determined based upon the data associated with previously received communications. Alternatively, an interface may be provided to an application administrator to allow configuration of particular thresholds with respect to individual anomaly types. In

some embodiments, thresholds by default may be dynamically derived unless specifically configured by an application administrator.

Anomalies are typically detected based upon a specific time period. Such a time period could be a particular fixed period (e.g., prior month, prior day, prior year, since security device's last reboot, etc.) and apply to all anomaly types. Alternatively, the time period for all anomaly types, or each anomaly type individually, may be configurable by an application administrator through an appropriate interface such as those discussed above. Some embodiments may support a fixed period default for all anomaly types, or each anomaly type individually, which may be overridden by application administrator configuration.

In one preferred embodiment, as depicted in FIG. 6, information from the risk profiles 642, 644, 646 generated by the collection engine is compared with the acquired thresholds for anomaly types of interest. Based upon these comparisons, a determination is made as to whether the received communication is anomalous, and if so, in what way (anomaly type) the communication is anomalous.

In one preferred embodiment, the stored risk profile associated with the received communication is aggregated with data associated with previously received communications of the same type. This newly aggregate data set is then used in analysis of subsequently received communications of that type.

If an anomaly is detected, an anomaly indicator signal is output. The outputted signal may include data identifying the anomaly type detected and the communication in which the anomaly was detected such as alert data 650. Various types of anomalies are discussed below with respect to e-mail application security. These types of anomalies may be detected using the specific detection approach discussed below or any of the aforementioned alternative anomaly detection approaches.

3. Action Engine

Based on the analysis, this component takes a decision of what sort of action needs to be triggered. Generally the action involves alerting the administrator of the ongoing unusual activity. An alert engine 660 performs this task by providing any appropriate notifications and/or initiating any appropriate corrective actions.

The outputted signal may trigger a further response in some embodiments; alternatively, the outputted signal may be the response. In one preferred embodiment, the outputted signal may be a notification to one or more designated recipient via one or more respective, specified delivery platform. For instance, the notification could be in the form of an e-mail message, a page, a facsimile, an SNMP alert, an SMS message, a WAP alert, OPSEC warning a voice phone call or other suitable message. Alternatively, such a notification could be triggered by the outputted signal.

Instead of or in addition to a notification, one or more corrective measures could be triggered by the outputted signal. Such corrective measures could include refusing acceptance of further communications from the source of the received communication, quarantining the communication, stripping the communication so that it can be safely handled by the application server, and/or throttling excessive numbers of incoming connections per second to levels manageable by internal application servers.

In one preferred embodiment, an interface may be provided that allows an application administrator to selectively configure a desired response and associate this configured response with a particular anomaly type such that when an anomaly of that type is detected the configured response occurs.

FIG. 4 depicts a flow chart in a typical anomaly detection process according to one preferred embodiment of the present invention. The process starts in step 410 by initializing various constraints of the process including the types of anomalies, thresholds for these types and time periods for which prior data is to be considered. This information may be configured interactively at initiation. In addition to, or instead of, the interactive configuration, previously stored configuration information may be loaded from the SDS.

The process continues at step 420 where anomaly definitional information is read (e.g., Incoming messages that have the same attachment within a 15 minute interval.). A determination is then made as to whether a new thread is needed; this determination is based upon the read the anomaly details (step not shown). In step 430, if a new thread is required, the thread is spun for processing in step 450. In step 440,

the process sleeps for a specified period of time before returning to step 420 to read information regarding an anomaly.

Once processing of the new thread commences in step 450, information needed to evaluate the anomaly is retrieved from appropriate locations in the SDS, manipulated if needed, and analyzed in step 460. A determination in step 470 occurs to detect an anomaly. In one preferred embodiment, this step uses predetermined threshold values to make the determination; such predetermined threshold values could be provided interactively or via a configuration file. If an anomaly is not detected, the process stops.

If an anomaly is detected, an anomaly indicator signal is output at step 480 which may result in a notification. The possible results of anomaly detection are discussed in more detail above with respect to the Action Engine.

The types of anomalies may vary depending upon the type and nature of the particular application server. The following discussion provides exemplary definitions of anomalies where e-mail is the application context in question. Anomalies similar, or identical, to these can be defined with respect to other application server types.

There are many potential anomaly types of interest in an e-mail system. The analysis is based on the collected data and dynamic rules for normality based on the historic audited data. In some embodiments, an application administrator can be provided with an interface for configuring predefined rules with respect to different anomaly types. FIG. 5 provides a sample screen for such an interface. The interface functionality may be provided via a Web server running on the security enhancement device or other suitable interface platform as discussed above.

In one preferred embodiment, the threshold value for the analysis for each anomaly is derived from an anomaly action table. The action for each anomaly is also taken from this table. The analysis identifies that some thing unusual has occurred and hands over to the action module. Enumerated below with respect to e-mail are anomalies of various types.

1. Messages from same IP Address – The point of collection for this anomaly is SMTPI/SMTPIS service. SMTPI/SMTPIS has information about the IP address

from which the messages originate. The IP address is stored in the SDS. The criterion for this anomaly is that the number of message for the given period from the same IP address should be greater than the threshold. Based on the level of threshold, suitable alert is generated.

2. Messages from same Address (MAIL FROM) – The point of collection for this anomaly is SMTPI/SMTPIS service. SMTPI/SMTPIS has information about the address (MAIL FROM) from which the messages originate. The determined address is stored in the SDS. The criterion for this anomaly is that the number of message for the given period with the same MAIL FROM address should be greater than the threshold. Based on the level of threshold, suitable alert is generated.
3. Messages having same Size – The point of collection for this anomaly is SMTPI/SMTPIS service. SMTPI/SMTPIS has information about the size of the messages. The size of the message is stored in the SDS. This size denotes the size of the message body and does not include the size of the headers. The criterion for this anomaly is that the number of message for the given period with a same size should be greater than the threshold. Based on the level of threshold, suitable alert is generated.
4. Messages having same Subject – The point of collection for this anomaly is SMTPI/SMTPIS service. SMTPI/SMTPIS has information about the subject line of the message. The subject line information for the message is stored in the SDS. The criterion for this anomaly is that the number of message for the given period with the same subject line should be greater than the threshold. Based on the level of threshold, suitable alert is generated.
5. Messages having same Attachment – The point of collection for this anomaly is the MIME Ripper Queue. The MIME Ripper Queue parses the actual message into the constituent MIME parts and stores the information in the SDS. A part of this information is the attachment file name. The criterion for this anomaly is that the number of message for the given period with same attachment name

should be greater than the threshold. Based on the level of threshold, suitable alert is generated.

6. Messages having same Attachment Extension - The point of collection for this anomaly is the MIME Ripper Queue. The MIME Ripper Queue parses the actual message into the constituent MIME parts and stores the information in the SDS. A part of this information is the attachment file extension. The criterion for this anomaly is that the number of message for the given period with same extension should be greater than the threshold. Based on the level of threshold, suitable alert is generated.
7. Messages having Viruses – This anomaly will be detected only if any of the anti-virus queues are enabled. The point of collection for this anomaly is the anti-virus Queue. The anti-virus Queue scans for any viruses on each individual MIME parts of the message. The scan details are stored in the SDS. A part of this information is the virus name. The criterion for this anomaly is that the number of message for the given period detected with viruses should be greater than the threshold. Based on the level of threshold, suitable alert is generated.
8. Messages having same Virus – This anomaly will be detected only if any of the anti-virus queues are enabled. The point of collection for this anomaly is the anti-virus Queue. The anti-virus Queue scans for any viruses on each individual MIME parts of the message. The scan details are entered into the SDS. A part of this information is the virus name. The criterion for this anomaly is that the number of message for the given period detected with same virus should be greater than the threshold. Based on the level of threshold, suitable alert is generated.

The table below depicts the fields in an anomaly table in one preferred embodiment using a relational database model for storing this information in the SDS.

Sl No.	Field Name	Data Type	Remarks
1.	anm_type	int	Primary key. Unique identifier for all anomalies. The list is given in next section.

Sl No.	Field Name	Data Type	Remarks
2.	anm_name	varchar	Name of the Anomaly (Tag for the UI to display)
3.	can_display	tinyint	Anomaly is displayable or not in UI. 0 – Do not display 1 – Display
4.	is_enabled	tinyint	Specifies if the anomaly is enabled or not 0 – Disabled 1 – Enabled
5.	anm_period	int	Time in minutes. This time specifies the period for the anomaly check.

The table below depicts the fields in an anomaly action table in one preferred embodiment using a relational database model for storing this information in the SDS.

Sl No.	Field Name	Data Type	Remarks
1.	anm_type	int	Foreign key from anomaly table.
2.	anm_thresh	int	This value specifies the threshold for a particular action to be taken.
3.	alert_type	int	This is foreign key from alert_type table. This value specifies the type of alert to be sent to the alert manager when this anomaly is detected.

Throughout this application, various publications may have been referenced. The disclosures of these publications in their entireties are hereby incorporated by reference into this application in order to more fully describe the state of the art to which this invention pertains.

The embodiments described above are given as illustrative examples only. It will be readily appreciated by those skilled in the art that many deviations may be made

from the specific embodiments disclosed in this specification without departing from the invention. Accordingly, the scope of the invention is to be determined by the claims below rather than being limited to the specifically described embodiments above.

5

What is claimed is:

1. An application layer security system, the system comprising:
   a) at least one application server system communication interface communicatively coupling the security system to one or more application server systems;
   b) a system data store capable of storing an electronic communication and accumulated data associated with received electronic communications; and
   c) a system processor in communication with the system data store and the at least one application server system communication interface, wherein the system processor comprises one or more processing elements and wherein the system processor:
      i) receives an electronic communication directed to or from a selected application server system;
      ii) applies one or more tests to the received electronic communication, wherein each of the one or more tests evaluates the received electronic communication for a particular security risk;
      iii) stores in the system data store a risk profile associated with the received electronic communication based upon the applied one or more tests;
      iv) determines whether an anomaly exists with respect to the received electronic communication based upon the stored risk profile and accumulated data associated with received electronic communications from the system data store; and
      v) outputs an anomaly indicator signal if an anomaly is determined to exist.
2. The system of claim 1, wherein the received electronic communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.
3. The system of claim 2, wherein the received electronic communication is an e-mail communication.

4. The system of claim 1, wherein each of the one or more tests applied by the system processor comprises intrusion detection, virus detection, spam detection or policy violation detection.
5. The system of claim 1, wherein the system processor applies a plurality of tests.
6. The system of claim 5, wherein the system processor applies each of the plurality of tests in a parallel fashion.
7. The system of claim 5, wherein the system processor applies each of the plurality of tests in a sequential fashion.
8. The system of claim 7, wherein the system data store comprises:

   i) a message data store capable of storing an electronic communication, and

   ii) a queue data store capable of storing a plurality of index queues; and

   wherein the system processor applies the plurality of tests in a sequential fashion by:

   1) storing the received electronic communication in the message data store;
   2) assigning a selected index to the stored electronic communication;
   3) executing a plurality of testing engines, wherein each of the testing engines has a test type and has an index queue in the queue data store associated with it, wherein at any given time at least two of the executing testing engines have differing test types, and wherein each of the testing engines:
      (a) monitors its associated index queue for a placed index;
      (b) retrieves the electronic communication associated with the placed index from the message data store; and
      (c) tests the retrieved electronic communication against a set of one or more criteria; and
   4) placing the selected index into the index queue associated with a first testing engine, wherein the first testing engine has a first test type; and
   5) placing the selected index into the index queue associated with a second testing engine, after the first testing engine performs its test upon the stored electronic communication associated with the selected index,

wherein the second testing engine has a second test type that differs from the first test type.

9. The system of claim 8, wherein the test type of each executing test engine is intrusion detection, virus detection, spam detection or policy violation detection.
10. The system of claim 1, wherein the system processor applies each of the one or more tests based upon configuration information stored in the system data store.
11. The system of claim 1, wherein the system processor determines whether an anomaly exists further based upon configuration information stored in the system data store.
12. The system of claim 11, wherein the configuration information comprises anomaly types, anomaly threshold information, anomaly time period information or anomaly response information.
13. The system of claim 1, wherein the system processor further derives one or more anomaly thresholds from the accumulated data associated with received electronic communications in the system data store.
14. The system of claim 1, wherein the system processor determines whether an anomaly exists by:
    1) determining a set of anomaly types of interest;
    2) for each of the anomaly types of interest in the determined set,
        (a) acquiring one or more anomaly thresholds associated with the respective anomaly type based at least in part upon accumulated data associated with received electronic communications from the system data store;
        (b) comparing information in the stored risk profile against at least one of the acquired one or more anomaly thresholds; and
        (c) determining whether an anomaly of the respective anomaly type exists with respect to the received electronic communication based upon the comparison.

15. The system of claim 14, wherein the system processor determines the set of anomaly types of interest by reading configuration information from the system data store.
16. The system of claim 14, wherein the system processor determines the set of anomaly types of interest based upon the received electronic communication.
17. The system of claim 14, wherein the system processor acquires the one or more anomaly thresholds by deriving at least one anomaly threshold from the accumulated data associated with received electronic communications.
18. The system of claim 17, wherein the derivation of the at least one anomaly threshold is further based upon a predetermined time period.
19. The system of claim 14, wherein the system processor acquires at least one anomaly threshold of the one or more anomaly thresholds by reading configuration information from the system data store.
20. The system of claim 1, wherein the anomaly indicator signal comprises a notification conveyed to an administrator.
21. The system of claim 20, wherein the notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or an SNMP alert.
22. The system of claim 20, wherein the anomaly indicator signal further comprises an anomaly type and wherein the notification conveyed to the administrator comprises the anomaly type.
23. The system of claim 1, wherein the anomaly indicator signal comprises an anomaly type.
24. The system of claim 1, wherein the system is disposed between a firewall system and one or more application server systems.
25. The system of claim 24, further comprising a firewall communication interface communicatively coupling the system to the firewall system, wherein the system processor receives the electronic communication directed to the selected application server system via the firewall communication interface.

26. The system of claim 1, wherein the system processor further forwards the received electronic communication to a destination indicated by the received electronic communication.
27. The system of claim 1, wherein the system processor further aggregates the stored risk profile with the accumulated data associated with received electronic communications and stores aggregated accumulated data in the system data store.
28. The system of claim 1, wherein the system processor further stores the received electronic communication in the system data store.
29. The system of claim 1, wherein the system processor further determines tests to apply to the received communication.
30. The system of claim 29, wherein the system processor determines test to be applied based upon configuration information stored in the system data store.
31. The system of claim 29, wherein the system processor determines test to be applied based upon characteristics of the received electronic communication.
32. The system of claim 1, wherein the system processor further provides an interface via which an administrator enters configuration information, receives configuration information from the interface and stores the received configuration information in the system data store.
33. The system of claim 32, wherein the system processor applies the one or more tests based upon the stored configuration information.
34. The system of claim 32, wherein the system processor determines whether an anomaly exists based upon the stored configuration information.
35. The system of claim 34, wherein the stored configuration information comprises anomaly types, anomaly threshold information, anomaly time period information or anomaly response information.
36. The system of claim 32, wherein the system processor provides the interface to the administrator via a Web server, an e-mail server, an automated voice recognition system or an SMS message server.
37. The system of claim 32, wherein the system processor further populates the interface with default values prior to providing it to the administrator.

38. The system of claim 1, wherein the system processor further takes a corrective measure responsive to the anomaly indicator signal.
39. The system of claim 38, wherein the corrective measure comprises conveying a notification to an administrator, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to levels manageable by internal application servers.
40. The system of claim 39, wherein the notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert.
41. The system of claim 1, wherein the one or more application server systems comprise e-mail server systems, Web server systems, FTP server systems, telnet server systems, GOPHER server systems or WAIS server systems.
42. The system of claim 41, wherein the one or more application server systems are e-mail server systems.
43. A method for enhancing application layer communication security, the method comprising the steps of:
    a) receiving an electronic communication directed to or from a selected application server system, wherein the received electronic communication is an application layer communication;
    b) applying one or more tests to the received electronic communication, wherein each of the one or more tests evaluates the received electronic communication for a particular security risk;
    c) determining whether an anomaly exists with respect to the received electronic communication based upon the applied one or more tests; and
    d) outputting an anomaly indicator signal if an anomaly is determined to exist.
44. The method of claim 43, wherein the received electronic communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

45. The method of claim 44, wherein the received electronic communication is an e-mail communication.

46. The method of claim 43, wherein the step of applying one or more tests comprises applying one or more of an intrusion detection test, a virus detection test, a spam detection test or a policy violation test.

47. The method of claim 43, wherein the step of applying one or more tests comprises sequentially applying a plurality of tests.

48. The method of claim 47, wherein the step of applying one or more tests comprises for each of the plurality of tests performing the steps of:

    i) selecting a test to perform,

    ii) performing the selected test on the received electronic communication, and

    iii) outputting a risk profile based upon the performed test.

49. The method of claim 48, and further comprising the step of receiving configuration information and wherein the step of selecting a test comprises selecting a test based upon the received configuration information.

50. The method of claim 48, wherein the step of selecting a test comprises selecting a test based upon a type associated with the received electronic communication.

51. The method of claim 43, and further comprising the step of receiving configuration information and wherein the step of determining whether an anomaly exists is further based upon the received configuration information.

52. The method of claim 51, wherein the received configuration information comprises anomaly types or anomaly response information.

53. The method of claim 52, and further comprising the step of receiving accumulated data associated with received communication and wherein the step of determining whether an anomaly exists comprises deriving anomaly threshold information from the received accumulated data.

54. The method of claim 52, wherein the received configuration information further comprises anomaly threshold information or anomaly time period information.

55. The method of claim 43, and further comprising the step of receiving accumulated data associated with received communication and wherein the step of determining

whether an anomaly exists is further based upon the received accumulated data associated with received communications.

56. The method of claim 43, and further comprising the step of taking a corrective measure responsive to the anomaly indicator signal.

57. Computer readable storage media storing instructions that upon execution by a system processor cause the system processor to provide application layer security, the media having stored instruction that cause the system processor to perform the steps comprising of:

a) receiving an electronic communication directed to or from a selected application server system, wherein the received electronic communication is an application layer communication;

b) applying one or more tests to the received electronic communication, wherein each of the one or more tests evaluates the received electronic communication for a particular security risk, thereby generating at least one risk profile associated with the electronic communication;

c) determining whether an anomaly exists with respect to the received electronic communication based upon the at least one risk profile; and

d) outputting an anomaly indicator signal if an anomaly is determined to exist.

58. The media of claim 57, wherein the instructions causing the system processor to receive the electronic communication comprise instructions causing the system processor to receive an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

59. The media of claim 57, wherein the instructions causing the system processor to receive the electronic communication comprise instructions causing the system processor to receive an e-mail communication.

60. The media of claim 57, wherein the instructions causing the system processor to apply one or more tests comprise instructions causing the system processor to apply one or more of an intrusion detection test, a virus detection test, a spam detection test or a policy violation test.

61. The media of claim 57, wherein the instructions causing the system processor to apply one or more tests comprise instructions causing the system processor to apply a plurality of tests.

62. The media of claim 61, wherein the instructions causing the system processor to apply a plurality of tests comprise instructions causing the system processor to:

    i) select a test to perform;

    ii) perform the selected test on the received electronic communication; and

    iii) outputting a risk profile based upon the performed test.

63. The media of claim 57, wherein the instructions causing the system processor to determine whether an anomaly exists comprises instruction causing the system processor to:

    i) determine a set of anomaly types of interest;

    ii) for each of the anomaly types of interest in the determined set,

        1) acquire one or more anomaly thresholds associated with the respective anomaly type;

        2) compare information in the at least one risk profile against at least one of the acquired one or more anomaly thresholds; and

        3) determine whether an anomaly of the respective anomaly type exists with respect to the received electronic communication based upon the comparison.

64. The media of claim 57, wherein the instructions causing the system processor to output the anomaly indicator signal comprise instructions causing the system to convey a notification to an administrator, wherein the notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or an SNMP alert.

65. An application layer security system, the system comprising:

    a) receiving means for receiving an application layer electronic communication;

    b) storing means for storing an electronic communication and accumulated data associated with received electronic communications;

c) assessment means for applying one or more tests to the received electronic communication, wherein each of the one or more tests evaluates the received electronic communication for a particular security risk, and for storing a risk profile in the storing means, wherein the risk profile was generated from applying the one or more tests to the received electronic communication;

d) anomaly determining means for determining whether an anomaly exists with respect to the received communication based upon the risk profile and accumulated data associated with the received electronic communications in the storing means; and

e) output means for outputting an anomaly indicator signal if an anomaly was determined to exist by the anomaly determining means.

66. A security system for interrogation of a communication transmitted over a communication network, the system comprising:

a) a communication interface communicatively coupling the system to a communication network;

b) a system data store comprising:

i) a message data store capable of storing a communication; and

ii) a queue data store capable of storing a plurality of index queues; and

c) a system processor in communication with the communication interface and the system data store, wherein the system processor comprises one or more processing elements and wherein the system processor:

i) receives a communication via the communication interface;

ii) stores the received message in the message data store;

iii) assigns a selected index to the stored communication;

iv) executes a plurality of interrogation engines, wherein each of the interrogation engines has a test type and has an index queue in the queue data store associated with it, and wherein each of the interrogation engines:

1) monitors its associated index queue for a placed index;

2) retrieves the communication associated with the placed index from the message data store;

3) assesses the retrieved communication against a set of one or more criteria related to the interrogation engine's test type; and

4) outputs an assessment indicator indicating results of assessing the retrieved communication with respect to the set of one or more criteria; and

v) places the selected index into the index queue associated with a first interrogation engine, wherein the first interrogation engine has a first test type; and

vi) responsive to the assessment indicator output by the first interrogation engine, places the selected index into the index queue associated with a second interrogation engine, wherein the second interrogation engine has a second test type that differs from the first test type.

67. The system of claim 66, wherein the received communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

68. The system of claim 67, wherein the received communication is an e-mail communication.

69. The system of claim 66, wherein the system processor assigns the selected index by retrieving a previously assigned index and incrementing it by a fixed amount.

70. The system of claim 66, wherein the system processor assigns the selected index based upon the received communication.

71. The system of claim 66, wherein the plurality of interrogation engines comprises one or more further interrogation engines each having a type differing from any interrogation engine that previously assessed the communication associated with the selected index and wherein the system processor, responsive to an assessment indicator output by an interrogation engine, places the selected index in an index queue associated with an interrogation engine having a type differing from any interrogation engine that previously assessed the communication associated with the selected index.

72. The system of claim 66, wherein each of the plurality of interrogation engines comprises intrusion detection, virus detection, spam detection or policy violation detection.
73. The system of claim 66, wherein the system processor adds an interrogation engine to the plurality of interrogation engines based upon the received communication.
74. The system of claim 66, wherein the system processor adds an interrogation engine to the plurality of interrogation engines based upon loading of existing interrogation engines in the plurality of interrogation engines.
75. The system of claim 66, wherein the assessment indicator output by each interrogation engine comprises a risk level associated with received communication with respect to the interrogation engine's test type.
76. The system of claim 66, wherein the assessment indicator output by each interrogation engine is output only if the interrogation engine's assessment of the received communication with respect to the set of one or more criteria determines that the received communication meets a threshold risk level.
77. The system of claim 76, wherein the assessment indicator signal comprises a notification conveyed to an administrator.
78. The system of claim 77, wherein the notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert.
79. The system of claim 77, wherein the assessment indicator further comprises the interrogation engine's test type for the interrogation engine outputting the assessment indicator and wherein the notification conveyed to the administrator comprises the interrogation engine's test type.
80. The system of claim 66, wherein the system processor deletes an interrogation engine from the plurality of interrogation engines based upon loading of existing interrogation engines in the plurality of interrogation engines.
81. The system of claim 66, wherein the system data store further comprises a configuration data store capable of storing configuration information.
82. The system of claim 81, wherein the system processor further updates configuration information based upon data accumulated from received communications.

83. The system of claim 81, wherein the system processor further provides an interface via which an administrator enters configuration information, receives configuration information from the interface and stores the received configuration information in the system data store.
84. The system of claim 83, wherein the system processor further populates the interface with one or more default values prior to providing it to the administrator.
85. The system of claim 83, wherein the system processor provides the interface to the administrator via a Web server, an e-mail server, an automated voice recognition system or an SMS message server.
86. The system of claim 81, wherein the system processor further determines the first interrogation engine based upon configuration information stored in the configuration data store.
87. The system of claim 81, wherein the system processor further determines the second interrogation engine based upon configuration information stored in the configuration data store.
88. The system of claim 81, wherein the system processor further determines the plurality of interrogation engines based upon configuration information stored in the configuration data store.
89. The system of claim 66, further comprising a firewall communication interface communicatively coupling the system to a firewall system, wherein the system processor receives the communication via the firewall communication interface.
90. The system of claim 66, further comprising a firewall communication interface communicatively coupling the system to a firewall system, wherein the received communication originates from a system connected to the communications network with a destination address external to the communications network.
91. The system of claim 66, wherein the received communication is directed to a system connected to the communications network.
92. The system of claim 66, wherein the received communication originates from a system connected to the communications network.

93. The system of claim 66, wherein the system processor further forwards the received communication to its destination.

94. The system of claim 93, wherein the system processor further determines the communication's destination based upon the communication.

95. The system of claim 66, wherein the system processor further outputs an overall indicator signal responsive to one or more assessment indicators respectively output from the plurality of interrogation engines.

96. The system of claim 66, wherein the system processor further takes a corrective measure responsive to one or more assessment indicators respectively output from the plurality of interrogation engines.

97. The system of claim 96, wherein the corrective measure comprises conveying a notification to an administrator, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to manageable levels for the communication network.

98. A method for interrogation of a communication transmitted over a communication network, the method comprising the steps of:

a) receiving a communication transmitted over a communication network;

b) assigning a selected index to the received communication;

c) executing a plurality of interrogation engines, wherein each of the interrogation engines has a test type and has an index queue associated with it, and wherein each of the interrogation engines performs the steps comprising of:

i) monitoring its associated index queue for a placed index;

ii) assessing the communication associated with the placed index against a set of one or more criteria related to the interrogation engine's type; and

iii) outputting an assessment indicator indicating results of assessing the communication associated with the placed index with respect to the set of one or more criteria;

d) placing the selected index into the index queue associated with a first interrogation engine, wherein the first interrogation engines has a first test type; and

e) responsive to the assessment indicator output by the first interrogation engine, placing the selected index into the index queue associated with a second interrogation engine, wherein the second interrogation engine has a second test type that differs from the first test type.

99. The method of claim 98, wherein the received communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

100. The method of claim 99, wherein the received communication is an e-mail communication.

101. The method of claim 98, wherein the plurality of interrogation engines comprises one or more further interrogation engines each having a type differing from any interrogation engine that previously assessed the communication associated with the selected index and further comprising the step of placing the selected index in an index queue associated with a subsequent interrogation engine having a type differing from any interrogation engine that previously assessed the communication associated with the selected index responsive to an assessment indicator output by an interrogation engine that previously assessed the communication.

102. The method of claim 98, wherein each of the plurality of interrogation engines comprises intrusion detection, virus detection, spam detection or policy violation detection.

103. The method of claim 98, and further comprising the step of adding an interrogation engine to the plurality of interrogation engines based upon the received communication.

104. The method of claim 98, and further comprising the step of adding an interrogation engine to the plurality of interrogation engines based upon loading of existing interrogation engines in the plurality of interrogation engines.

105. The method of claim 98, wherein the assessment indicator output by each interrogation engine is output only if the interrogation engine's assessment of the received communication with respect to the set of one or more criteria determines that the received communication meets a threshold risk level.

106. The method of claim 105, wherein the assessment indicator signal comprises a notification conveyed to an administrator.

107. The method of claim 98, and further comprising the step of deleting an interrogation engine from the plurality of interrogation engines based upon loading of existing interrogation engines in the plurality of interrogation engines.

108. The method of claim 98, and further comprising the step of taking a corrective measure responsive to one or more assessment indicators respectively output from the plurality of interrogation engines.

109. The method of claim 108, wherein the corrective measure comprises conveying a notification to an administrator, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to manageable levels for the communication network.

110. Computer readable storage media storing instructions that upon execution by a system processor cause the system processor to interrogate a communication transmitted over a communication network, the media having stored instruction that cause the system processor to perform the steps comprising of:

a) receiving a communication transmitted over a communication network;

b) assigning a selected index to the received communication;

c) executing a plurality of interrogation engines, wherein each of the interrogation engines has a test type and has an index queue associated with it, and wherein each of the interrogation engines performs the steps comprising of:

i) monitoring its associated index queue for a placed index;

ii) assessing the communication associated with the placed index against a set of one or more criteria related to the interrogation engine's type; and

iii) outputting an assessment indicator indicating results of assessing the communication associated with the placed index with respect to the set of one or more criteria;

d) placing the selected index into the index queue associated with a first interrogation engine, wherein the first interrogation engines has a first test type; and

e) responsive to the assessment indicator output by the first interrogation engine, placing the selected index into the index queue associated with a second interrogation engine, wherein the second interrogation engine has a second test type that differs from the first test type.

111. The media of claim 110, wherein the instructions causing the system processor to receive the communication comprise instructions causing the system processor to receive an e-mail communication.

112. The media of claim 110, wherein the plurality of interrogation engines comprises one or more further interrogation engines each having a type differing from any interrogation engine that previously assessed the communication associated with the selected index and further comprising instructions causing the system processor to perform the step of placing the selected index in an index queue associated with a subsequent interrogation engine having a type differing from any interrogation engine that previously assessed the communication associated with the selected index responsive to an assessment indicator output by an interrogation engine that previously assessed the communication.

113. The media of claim 110, wherein each of the plurality of interrogation engines comprises intrusion detection, virus detection, spam detection or policy violation detection.

114. The media of claim 110, and further comprising instructions causing the system processor to perform the step of adding an interrogation engine to the plurality of interrogation engines based upon loading of existing interrogation engines in the plurality of interrogation engines.

115. The media of claim 110, wherein the assessment indicator output by each interrogation engine is output only if the interrogation engine's assessment of the received communication with respect to the set of one or more criteria determines that the received communication meets a threshold risk level.

116. The media of claim 110, and further comprising instructions causing the system processor to perform the step of deleting an interrogation engine from the plurality of interrogation engines based upon loading of existing interrogation engines in the plurality of interrogation engines.

117. The media of claim 110, and further comprising instructions causing the system processor to perform the step of taking a corrective measure responsive to one or more assessment indicators respectively output from the plurality of interrogation engines.

118. A security system for interrogation of a communication transmitted over a communication network, the system comprising:

a) receiving means for receiving a communication transmitted over a communication network;

b) storing means for storing a received communication and a plurality of index queues;

c) assignment means for assigning a selected index to a stored communication;

d) interrogation engine management means for executes a plurality of interrogation engines, wherein each of the interrogation engines has a test type and has an index queue in the queue data store associated with it, and wherein each of the interrogation engines:

1) monitors its associated index queue for a placed index;

2) retrieves the communication associated with the placed index from the message data store;

3) assesses the retrieved communication against a set of one or more criteria related to the interrogation engine's test type; and

4) outputs an assessment indicator indicating results of assessing the retrieved communication with respect to the set of one or more criteria; and

e) index placement means for placing the selected index in a index queue associated with an interrogation engine, wherein the index placement means places the selected index into the index queue of a first interrogation engine responsive to assignment of the selected index by the index assignment means and wherein the index placement means places the selected index into the index queue associated with an interrogation engine having a type differing from any interrogation engine that previously assessed the communication associated with the selected index responsive to an assessment indicator output by an interrogation engine that previously assessed the communication.

119. A system for detecting an anomalous communication transmitted over a communications network, the system comprising:

a) an interface coupling the system with the communications network;

b) a system data store capable of storing data associated with communications transmitted over the communications network and information associated with one or more responses to be initiated if an anomaly is detected;

c) a system processor in communication with the interface and the data store, wherein the system processor comprises one or more processing elements and wherein the system processor executes:

i) a collection engine that:

1) receives a communication via the interface; and

2) generates data associated with the received communication by applying one or more tests to the received communication;

ii) an analysis engine that detects whether an anomaly exists with respect to the received communication based upon the data generated by the collection engine and data associated with previously received communications from the system data store; and

iii) an action engine that initiates a predetermined response from the system data store if an anomaly was detected by the analysis engine.

120. The system of claim 119, wherein the received communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

121. The system of claim 120, wherein the received communication is an e-mail communication.

122. The system of claim 119, wherein each of the one or more tests applied by the collection engine comprises intrusion detection, virus detection, spam detection or policy violation detection.

123. The system of claim 119, wherein the collection engine applies a plurality of tests.

124. The system of claim 123, wherein the collection engine applies each of the plurality of tests in a parallel fashion.

125. The system of claim 123, wherein the collection engine applies each of the plurality of tests in a sequential fashion.

126. The system of claim 119, wherein the system data store stores configuration information and wherein the collection engine applies each of the one or more tests based upon configuration information stored in the system data store.

127. The system of claim 119, wherein the analysis engine detects whether an anomaly exists further based upon configuration information stored in the system data store.

128. The system of claim 127, wherein the configuration information comprises anomaly types, anomaly threshold information, anomaly time period information or anomaly response information.

129. The system of claim 119, wherein the analysis engine further derives one or more anomaly thresholds from the accumulated data associated with received communications in the system data store and wherein the analysis engine detects whether an anomaly exists further based upon the derived one or more anomaly thresholds.

130. The system of claim 119, wherein the analysis engine detects whether an anomaly exists by:

1) determining a set of anomaly types of interest;

2) for each of the anomaly types of interest in the determined set,

(a) acquiring one or more anomaly thresholds associated with the respective anomaly type based at least in part upon accumulated data associated with received communications from the system data store;

(b) comparing information in the stored risk profile against at least one of the acquired one or more anomaly thresholds; and

(c) determining whether an anomaly of the respective anomaly type exists with respect to the received communication based upon the comparison.

131. The system of claim 130, wherein the system data store stores configuration information and wherein the analysis engine determines the set of anomaly types of interest by reading configuration information from the system data store.

132. The system of claim 130, wherein the analysis engine determines the set of anomaly types of interest based upon the received communication.

133. The system of claim 130, wherein the analysis engine acquires at least one of the one or more anomaly thresholds by deriving the at least one anomaly threshold from the accumulated data associated with previously received communications.

134. The system of claim 133, wherein the derivation of the at least one anomaly threshold is further based upon a predetermined time period.

135. The system of claim 130, wherein the system data store stores configuration information and wherein the analysis engine acquires at least one of the one or more anomaly threshold by reading configuration information from the system data store.

136. The system of claim 119, wherein the action engine's initiated predetermined response is based upon an anomaly type associated with an anomaly detected by the analysis engine.

137. The system of claim 119, wherein the action engine's initiated predetermined response comprises conveying a notification to an administrator, refusing

acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to manageable levels.

138. The system of claim 137, wherein the action engine's initiated predetermined response comprises conveying a notification to an administrator and wherein the notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert.

139. The system of claim 119, wherein the system processor further aggregates the data generated by the collection engine with the accumulated data associated with previously received communications and stores aggregated accumulated data in the system data store.

140. The system of claim 119, wherein the system processor further provides an interface via which an administrator enters configuration information, receives configuration information from the interface and stores the received configuration information in the system data store.

141. The system of claim 140, wherein the collection engine applies the one or more tests based upon the stored configuration information.

142. The system of claim 140, wherein the analysis engine detects whether an anomaly exists based upon the stored configuration information.

143. The system of claim 142, wherein the stored configuration information comprises anomaly types, anomaly threshold information, anomaly time period information or anomaly response information.

144. The system of claim 140, wherein the system processor provides the interface to the administrator via a Web server, an e-mail server, a automated voice recognition system or an SMS message server.

145. The system of claim 140, wherein the system processor further populates the interface with default values prior to providing it to the administrator.

146. A method for detecting an anomalous communication transmitted over a communication network, the method comprising the steps of:

a) receiving a communication transmitted over a communication network;

b) applying one or more tests to the received communication to generate data associated with the received communication;

c) acquiring data associated with one or more previously received communications;

d) detecting whether an anomaly exists with respect to the received communication based upon the generated data and acquired data; and

e) initiating a predetermined response if an anomaly was detected.

147. The method of claim 146, wherein the received communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

148. The method of claim 147, wherein the received communication is an e-mail communication.

149. The method of claim 146, wherein each of the one or more tests applied by the collection engine comprises intrusion detection, virus detection, spam detection or policy violation detection.

150. The method of claim 146, wherein the step of applying one or more tests comprises applying a plurality of tests.

151. The method of claim 146, and further comprising the step of deriving one or more anomaly thresholds from the acquired data and wherein the step of detecting whether an anomaly exists further bases detecting whether an anomaly exists upon the derived one or more anomaly thresholds.

152. The method of claim 146, wherein the step of detecting whether an anomaly exists comprises:

a) determining a set of anomaly types of interest;

b) for each of the anomaly types of interest in the determined set,

i) acquiring one or more anomaly thresholds associated with the respective anomaly type based at least in part upon the acquired data associated with one or more previously received communications;

ii) comparing information in the stored risk profile against at least one of the acquired one or more anomaly thresholds; and

iii) determining whether an anomaly of the respective anomaly type exists with respect to the received communication based upon the comparison.

153. The method of claim 152, wherein the step of determining a set of anomaly types of interest comprises reading a configuration file.

154. The method of claim 152, wherein the step of determining a set of anomaly types of interest determines the set based upon the received communication.

155. The method of claim 152, wherein the step of acquiring one or more anomaly thresholds comprises the step of deriving at least one anomaly threshold from the acquired data associated with one or more previously received communications.

156. The method of claim 146, wherein the initiated predetermined response is based upon an anomaly type associated with a detected anomaly.

157. The method of claim 146, wherein the initiated predetermined response comprises conveying a notification to an administrator, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to manageable levels.

158. The method of claim 157, wherein the initiated predetermined response comprises conveying a notification to an administrator and wherein the notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert.

159. Computer readable storage media storing instructions that upon execution by a system processor cause the system processor to detect an anomalous communication transmitted over a communication network, the media having stored instruction that cause the system processor to perform the steps comprising of:

a) receiving a communication transmitted over a communication network;

b) applying one or more tests to the received communication to generate data associated with the received communication;

c) acquiring data associated with one or more previously received communications;

d) detecting whether an anomaly exists with respect to the received communication based upon the generated data and acquired data; and

e) initiating a predetermined response if an anomaly was detected.

160. The media of claim 159, wherein the instructions causing the system processor to receive the communication comprise instructions causing the system processor to receive an e-mail communication.

161. The media of claim 159, wherein the instructions causing the system processor to apply one or more tests comprise instructions causing the system processor to apply one or more of an intrusion detection test, a virus detection test, a spam detection test or a policy violation test.

162. The media of claim 159, wherein the instructions causing the system processor to detect whether an anomaly exists comprise instructions causing the system processor to perform the steps comprising of:

a) determining a set of anomaly types of interest;

b) for each of the anomaly types of interest in the determined set,

i) acquiring one or more anomaly thresholds associated with the respective anomaly type based at least in part upon the acquired data associated with one or more previously received communications;

ii) comparing information in the stored risk profile against at least one of the acquired one or more anomaly thresholds; and

iii) determining whether an anomaly of the respective anomaly type exists with respect to the received communication based upon the comparison.

163. The media of claim 159, wherein the initiated predetermined response is based upon an anomaly type associated with a detected anomaly.

164. The media of claim 159, wherein the initiated predetermined response comprises conveying a notification to an administrator, refusing acceptance of

further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to manageable levels.

165. A system for detecting an anomalous communication transmitted over a communications network, the system comprising:

a) storing means for storing data associated with communications transmitted over the communications network and information associated with one or more responses to be initiated if an anomaly is detected;

b) collection means for receiving a communication transmitted over a communications network and for generating data associated with the received communication by applying one or more tests to the received communication;

c) analysis means for detecting whether an anomaly exists with respect to the received communication based upon the data generated by the collection means and data associated with previously received communications from the storing means; and

d) action means for initiating a predetermined response from the storing means if an anomaly was detected by the analysis means.

166. A management system for generating and distributing threat detection rules to application layer security systems, the system comprising:

a) a communication interface adapted to allow communication between the management system and at least one application layer security system;

b) a system data store comprising one or more data storage elements, wherein the system data store is capable of storing:

i) one or more sets of threat management goals; and

ii) threat information; and

c) a system processor in communication with the communication interface and the system data store, wherein the system processor comprises one or more processing elements and the one or more processing elements are programmed or adapted to:

i) receive threat information from one or more sources;

ii) reduce the received threat information into a canonical form;

iii) extract features from the reduced threat information;

iv) generate a rule set of one or more threat rules based upon the extracted features and a goal set of one or more threat management goals in the system data store; and

v) transmit the generated rule set to at least one of the plurality of application layer security systems.

167. The system of claim 166, wherein the system data store is further capable of storing one or more sets of test data, wherein the system processor is further programmed or adapted to evaluate the generated rule set against one or more sets of test data in the system data store and to refine the rule set if the evaluation of the rule set fails to satisfy a predetermined confidence level.

168. The system of claim 167, wherein the system processor evaluates the generated rule set against one or more sets of test data based upon the goal set used to generate the rule set.

169. The system of claim 168, wherein the second goal set comprises one or more values of a type selected from the group of effectiveness values, accuracy values, efficiency values and false positive values.

170. The system of claim 167, wherein the system processor evaluates the generated rule set against one or more sets of test data based upon a second goal set of one or more goals from the system data store that differs from the goal set used to generate the rule set.

171. The system of claim 170, wherein the second goal set comprises one or more values of a type selected from the group of effectiveness values, accuracy values, efficiency values and false positive values.

172. The system of claim 166, wherein the goal set comprises one or more attributes selected from the group consisting of effectiveness values, accuracy values, efficiency values and false positive values.

173. The system of claim 166, wherein the goal set comprises one or more undesirable message types, wherein each undesirable message type is selected from the group consisting of business email, personal email, chain letters, adult language, porn, web product offerings, newsletters, mailing lists, Trojans, worms and viruses.

174. The system of claim 173, wherein associated with each undesirable message type is a value corresponding to a level of undesirability for the message type.

175. The system of claim 166, wherein the system processor is further programmed or adapted to select the goal set from the system data store.

176. The system of claim 175, wherein the system processor's programming or adaptation to select the goal set includes programming or adaptation to select the goal set based at least in part upon a selected application layer security system from the plurality of application layer security systems.

177. The system of claim 176, wherein the system processor transmits the generated rule set to at least the selected application layer security system.

178. The system of claim 166, wherein the system processor receives threat information from a selected application layer security system via the communication interface.

179. The system of claim 166, wherein the system processor receives threat information from a spam database, a virus information database, an intrusion information database or combinations thereof.

180. The system of claim 179, wherein the system processor receives further threat information from a selected application layer security system via the communication interface.

181. The system of claim 166, wherein the system processor extracts features that each correspond to an interrogation type available on at least one of the plurality of application layer security systems.

182. The system of claim 166, wherein the system processor extracts features by applying one or more regular expression filters.

183. The system of claim 166, wherein the system processor is further programmed or adapted to receive at least one rules and policy application programming interface.

184. The system of claim 183, wherein the system processor generates the rule set based upon the at least one rules and policy application programming interface.

185. A method for managing threat information, the method comprising the steps of:

a) receiving threat information from one or more sources selected from the group consisting of application layer security systems, spam databases, a virus information databases, and intrusion information databases;

b) reducing the received threat information into a canonical form;

c) extracting features from the reduced threat information by applying one or more regular expressions;

d) selecting a goal set of one or more threat management goals based at least in part upon a selected application layer security system from the plurality of application layer security systems, wherein the goal set comprises one or more values of a type selected from the group of effectiveness values, accuracy values, efficiency values and false positive values;

e) generating a candidate rule set of one or more threat rules based upon the extracted features and the goal set;

f) testing the candidate rule set against one or more sets of test data;

g) refining the candidate rule set if the evaluation of the rule set fails to satisfy a predetermined confidence level; and

h) transmitting the candidate or refined rule set to at least one application layer security system.

186. A management system for generating and distributing threat detection rules to application layer security systems, the system comprising:

a) communication means for receiving threat information from one or more sources selected from the group consisting of application layer security systems, spam databases, a virus information databases, and intrusion information

databases and for transmitting a generated rule set to at least one application layer security system;

b) rule generation means for:

i) reducing threat information received by the communication means into a canonical form;

ii) extracting features from the reduced threat information by applying one or more regular expressions;

iii) selecting a goal set of one or more threat management goals based at least in part upon a selected application layer security system from the plurality of application layer security systems, wherein the goal set comprises one or more values of a type selected from the group of effectiveness values, accuracy values, efficiency values and false positive values;

iv) generating a candidate rule set of one or more threat rules based upon the extracted features and the goal set;

v) testing the candidate rule set against one or more sets of test data;

vi) refining the candidate rule set if the evaluation of the rule set fails to satisfy a predetermined confidence level; and

vii) providing the candidate or refined rule set to the communication means.

187. An application layer security system, the system comprising:

a) at least one application server system communication interface communicatively coupling the security system to a communication network allowing communication with one or more other application-layer security systems and a threat management system;

b) a system data store capable of storing an electronic communication and accumulated data associated with received electronic communications; and

c) a system processor in communication with the system data store and the at least one application server system communication interface, wherein the system processor comprises one or more processing elements and wherein the system processor:

i) receives an electronic communication directed to or from a selected application server system;

ii) applies one or more tests to the received electronic communication, wherein each of the one or more tests evaluates the received electronic communication for a particular security risk;

iii) stores in the system data store a risk profile associated with the received electronic communication based upon the applied one or more tests; and

iv) outputs information based upon the stored risk profile to a second application layer security system, a threat management center, a threat pushback system or a combination thereof.

188. The system of claim 187, wherein the received electronic communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

189. The system of claim 188, wherein the received electronic communication is an e-mail communication.

190. The system of claim 187, wherein each of the one or more tests applied by the system processor comprises intrusion detection, virus detection, spam detection or policy violation detection.

191. The system of claim 187, wherein the system processor applies a plurality of tests.

192. The system of claim 191, wherein the system processor applies each of the plurality of tests in a parallel fashion.

193. The system of claim 191, wherein the system processor applies each of the plurality of tests in a sequential fashion.

194. The system of claim 187, wherein the system processor applies each of the one or more tests based upon configuration information stored in the system data store.

195. The system of claim 187, wherein the system processor outputs the information in the form of an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or an SMNP alert.

196. The system of claim 187, wherein the system processor outputs the information to a threat management center and to a second application layer security system, a threat pushback system or a combination thereof.

197. The system of claim 187, wherein the one or more tests applied by the system processor comprise anomaly detection.

198. The system of claim 187, wherein an application layer security system can broadcast a message to at least one other application layer security system.

199. The system of claim 187, and further comprising a central threat management system, wherein the centralized threat management system coordinates threat information among multiple nodes.

200. The system of claim 187, wherein each application layer security system is a trusted application layer security system.

201. The system of claim 187, wherein the system processor outputs information comprising threat statistics.

202. The system of claim 201, wherein the threat statistics include information about the frequency of certain threat types.

203. The system of claim 187, wherein the system processor outputs information comprising information corresponding to a specific identified threat, one or more whitelists, traffic pattern data, or combinations thereof.

204. A method for identifying a back trail for an electronic communications, the method comprising the steps of:

a) receiving an electronic communication comprising a destination address;

b) identifying one or more addresses of computer systems in a path from a source to the destination address;

c) analyzing authenticity of at least one of the identified computer system addresses; and

d) outputting one or more computer system addresses analyzed to be authentic.

205. The method of claim 204, wherein the received electronic communication comprises a header that includes the destination address and wherein the step of

identifying one or more addresses comprises the steps of parsing the header for computer system addresses.

206. The method of claim 205, wherein the step of analyzing authenticity comprises identifying inconsistencies in the header.

207. The method of claim 206, wherein the step of identifying inconsistencies comprises identifying inconsistencies in date information, server version information, receive from information, recipient information or combinations thereof.

208. The method of claim 205, wherein the step of analyzing authenticity comprises identifying inconsistencies between the header, transfer information associated with the communication, content of the communication, or combinations thereof.

209. The method of claim 204, wherein the step of analyzing authenticity comprises analyzing authenticity of each of the identified computer system addresses.

210. The method of claim 204, wherein the step of analyzing authenticity comprises accessing an external database of address verification information.

211. The method of claim 210, wherein the external database comprises IP addresses or ASN numbers of known originators of spam.

212. The method of claim 210, wherein the address verification information comprises IP addresses, ASN numbers, host names, domain names, e-mail addresses or combinations thereof.

213. The method of claim 204, wherein the step of analyzing authenticity comprises performing a DNS lookup based upon the at least one identified computer system addresses.

214. The method of claim 204, wherein the step of analyzing authenticity comprises assigning a confidence level to each of the at least one identified computer system addresses.

215. The method of claim 214, wherein the step of assigning a confidence level to each of the at least one identified computer system addresses comprises assigning a confidence level to each of the identified computer system addresses.

216. The method of claim 214, wherein the step of analyzing authenticity further comprises comparing the assigned confidence level with a predetermined threshold.

217. The method of claim 214, wherein the step of outputting further comprises outputting the confidence level associated with each computer system address verified as authentic.

218. The method of claim 204, and further comprising the step of outputting computer system addresses not analyzed to be authentic.

219. A system for identifying a back trail for an electronic communications, the system comprising:

a) a interface adapted to link the system with a communication network;

b) a system data store capable of storing one or more electronic communications, data associated therewith, configuration data or combinations thereof; and

c) a system processor in communication with the interface and the system data store and comprising one or more processing elements, wherein the one or more processing elements are programmed or adapted to:

i) receive an communication via the interface, wherein the received communication comprises a header having a destination address;

ii) store the received communication in the system data store;

iii) parse the header of the received communication for one or more addresses of computer systems in a path between a source and the destination address;

iv) perform a plurality of tests on each of the one or more addresses to assign a confidence value to each address;

v) determine whether each address is valid by comparing its assigned confidence value to a predetermined threshold in the system data store; and

vi) output each address determined valid.

220. The system of claim 219, wherein the system processor outputs each address determined valid with a first indicator.

221. The system of claim 220, wherein the system processor is further programmed or adapted to output each address not determined valid with a second indicator.

222. A system for identifying a back trail for an electronic communication, the system comprising:

a) interface means for receiving communications transmitted over a communication network;

b) storage means for storing at least one or more received communications, data associated therewith, configuration data or combinations thereof;

c) identification means for identifying from a received communication one or more addresses along a path between a source of a received communication and a destination of the received communication;

d) verification means for verifying each address identified by the identification means comprising means for assigning a confidence value to each address and means for comparing the confidence value assigned to each address with a predetermined threshold in the storage means; and

e) means for outputting verified addresses.

223. Computer readable media storing instruction that upon execution by a system processor cause the system processor to identify a back trail for an electronic communication by performing the steps comprising of:

a) receiving an communication via the interface, wherein the received communication comprises a header having a destination address;

b) storing the received communication in the system data store;

c) parsing the header of the received communication for one or more addresses of computer systems in a path between a source and the destination address;

d) performing a plurality of tests on each of the one or more addresses to assign a confidence value to each address;

e) determining whether each address is valid by comparing its assigned confidence value to a predetermined threshold in the system data store; and

f) outputting each address determined valid.

224. A threat push-back system within, or in communication with, an application layer security system, a threat management center or an application client, the system comprising:

a) an interface communicatively coupling the system to a communication network;

b) a system data store capable of storing an electronic communication and accumulated data associated with one or more received electronic communications; and

c) a system processor in communication with the system data store and the interface, wherein the system processor comprises one or more processing elements and wherein the system processor:

i) receives a communication via the interface;

ii) generates a threat profile associated with the received communication;

iii) stores in the system data store the generated threat profile associated with the received communication;

iv) compares the generated threat profile with threat configuration information;

v) if the comparison indicates the received communication represent a threat,

1) determines one or more computer addresses in a back path of the received communication; and

2) outputs information based upon the stored threat profile to one or more upstream computers associated with one or more of the determined addresses.

225. The system of claim 224, wherein the system processor generates the threat profile by applying one or more tests to the received communication, wherein each of the one or more tests evaluates the received communication for a particular security risk.

226. The system of claim 225, wherein the system processor generates the threat profile by further determining whether an anomaly exists with respect to the received communication based upon the applied one or more tests and accumulated data associated with received communications from the system data store.

227. The system of claim 224, wherein the system processor further provides an interface for identifying threat types, receives threat type information from the provided interface and updates threat configuration information in the system data store based upon the received threat type information.

228. The system of claim 227, wherein the system processor provides an interface for identifying threat types comprising of business email, personal email, chain letters, adult language, porn, web product offerings, newsletters, mailing lists, worms, virus, server attack or combinations thereof.

229. The system of claim 227, wherein the system processor provides an interface allowing identification of threat types and specification of a weight associated for each identified threat type.

230. The system of claim 224, wherein the system processor compares the generated threat profile with threat configuration information by calculating a threat value from the threat profile and determining whether the threat value satisfies a predetermined threat condition.

231. The system of claim 230, wherein the system processor calculates the threat value by retrieving weights associate with threat types in the configuration data and threat type values associated with the respective threat types from the threat profile and generating a weighted sum based upon the retrieved weights and threat type values.

232. The system of claim 231, wherein the system processor calculates the threat value by selecting threat types for retrieval of weights and values based upon the predetermined threat condition.

233. The system of claim 224, wherein the system processor determines one or more back path computer addresses by at least parsing a header associated with the received communication for computer system addresses to identify computer addresses.

234. The system of claim 233, wherein the system processor determines one or more back path computer addresses by at least analyzing authenticity of at least one of the identified computer system addresses.

235. The system of claim 224, wherein the system processor is further programmed or adapted to map determined addresses into upstream computers by reference to a database of security.

236. The system of claim 224, wherein the system processor outputs information based upon the stored threat profile to one or more upstream threat management computers, one or more upstream application layer security systems or combinations thereof.

237. The system of claim 224, wherein the system processor outputs information to one or more upstream computers one or more threat types, one or more threat source identifiers, one or more threat details, one or more response details or combinations thereof.

238. The system of claim 237, wherein the system processor outputs information to one or more upstream computers one or more source identifiers, wherein each source identifier is an e-mail address, a domain name, a host name, an IP address or combinations thereof.

239. The system of claim 238, wherein each source identifier includes an associated confidence value.

240. The system of claim 224, wherein the system processor outputs information according to a standardized threat information protocol.

241. The system of claim 240, wherein the system processor outputs information according to a protocol selected from the group consisting of SNMP, Intrusion Detection Message Exchange Format (IDMEF), and Intrusion Detection Exchange Protocol (IDXP).

242. The system of claim 224, wherein the system processor is further programmed or adapted to output a threat notification if the comparison indicates the received communication represent a threat.

243. The system of claim 242, wherein the system processor outputs the threat notification to one or more computer systems.

244. The system of claim 242, wherein the system processor outputs the threat notification as an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert, an SNMP alert or combinations thereof.

245. The system of claim 224, wherein the system processor is further programmed or adapted to take a corrective measure if the comparison indicates the received communication represents a threat.

246. The system of claim 245, wherein the corrective measure comprises conveying a notification to one or more users, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to levels manageable by internal application servers.

247. The system of claim 246, wherein the corrective measure comprises conveying a notification to one or more users and wherein each notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert.

248. A computer implemented threat push-back method, the method comprising the steps of:

a) providing an interface for establishing configuration information regarding one or more threat types, wherein configuration information comprises threat types of interest and weights associated therewith;

b) receiving a communication;

c) generating a threat profile associated with the received communication by applying one or more tests to the received communication, wherein each of the one or more tests evaluates the received communication for a particular security risk;

d) comparing the generated threat profile with the configuration information by calculating a threat value from the threat profile and determining whether the threat value satisfies a predetermined threat condition; and

e) if the comparison indicates the received communication represents a threat,

i) determining one or more computer addresses in a back path of the received communication;

ii) analyzing authenticity of at least one of the determined computer system addresses;

iii) taking a corrective measure, wherein the corrective measure comprises conveying a notification to one or more users, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to levels manageable by internal application servers, wherein each notification conveyed comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert; and

iv) outputting information based upon the stored threat profile to one or more upstream application layer security system, one or more threat management system, or combinations thereof, wherein each upstream application layer security system or threat management system is associated with one or more of the determined addresses.

249. Computer readable media storing instructions that upon execution by a system processor cause the system processor to identify and push-back threat information upstream of an identified threat by performing the steps comprising of:

a) providing an interface for establishing configuration information regarding one or more threat types, wherein configuration information comprises threat types of interest and weights associated therewith;

b) receiving a communication;

c) generating a threat profile associated with the received communication by applying one or more tests to the received communication, wherein each of the one or more tests evaluates the received communication for a particular security risk;

d) comparing the generated threat profile with the configuration information by calculating a threat value from the threat profile and determining whether the threat value satisfies a predetermined threat condition; and

e) if the comparison indicates the received communication represents a threat,

i) determining one or more computer addresses in a back path of the received communication;

ii) analyzing authenticity of at least one of the determined computer system addresses;

iii) outputting a threat notification to one or more users, wherein each outputted threat notification comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert, or an SNMP alert; and

iv) outputting information based upon the stored threat profile to one or more upstream application layer security system, one or more threat management system, or combinations thereof, wherein each upstream application layer security system or threat management system is associated with one or more of the determined addresses.

250. A threat push-back system within, or in communication with, an application layer security system, a threat management center or an application client, the system comprising:

a) configuration means for establishing configuration information regarding one or more threat types, wherein configuration information comprises threat types of interest and weights associated therewith;

b) receiving means for receiving an electronic communication;

c) means for generating a threat profile associated with the received communication by applying one or more tests to the received communication, wherein each of the one or more tests evaluates the received communication for a particular security risk;

d) means for comparing the generated threat profile with the configuration information by calculating a threat value from the threat profile and determining whether the threat value satisfies a predetermined threat condition; and

e) back path identification means for determining one or more computer addresses in a back path of the received communication;

f) authentication means for authenticating at least one of the one or more computer addresses from the back path identification means;

g) output means for outputting information based upon the stored threat profile to one or more upstream application layer security system, one or more threat management system, or combinations thereof, wherein each upstream application layer security system or threat management system is associated with one or more of the determined addresses; and

a) means for taking a corrective measure, wherein the corrective measure comprises conveying a notification to one or more users, refusing acceptance of further communications from the source of the received communication, quarantine of the received communication, stripping the received communication of identified content, or throttling excessive numbers of incoming connections per second to levels manageable by internal application servers, wherein each notification conveyed comprises an e-mail message, a page, a facsimile, an telephone call, an SMS message, a WAP alert or SNMP alert.

251. A system for detecting an unsolicited communication transmitted over a communications network, the system comprising:

a) an interface coupling the system with the communications network;

b) a system data store capable of storing data associated with communications transmitted over the communications network and one or more whitelists;

c) a system processor in communication with the interface and the system data store, wherein the system processor comprises one or more processing elements and wherein the system processor:

i) receive a communication via the interface;

ii) compares the communication to at least one whitelist; and

iii) modifies at least one whitelist based on the communication.

252. The system of claim 251, wherein the system processor is programmed or adapted to modify the at least one whitelist by updating the at least one whitelist with data derived from the received communication.

253. The system of claim 251, wherein the system processor is further programmed or adapted to modify the at least one whitelist by updating the at least one whitelist with data derived from inbound or outbound communication traffic patterns.

254. The system of claim 253, wherein the system processor modifies the at least one whitelist by updating the at least one whitelist with data derived from inbound and outbound communication traffic patterns.

255. The system of claim 251, wherein the system processor is programmed or adapted to modify the at least one whitelist by adding an entry to the at least one whitelist corresponding to a destination address associated with the received communication.

256. The system of claim 251, wherein the system processor is further programmed or adapted to assign a confidence level to received communications.

257. The system of claim 256, wherein the system processor is further programmed or adapted to forward a communication with an indication of its confidence level.

258. The system of claim 251, wherein the communication is transmitted or received over the Internet.

259. The system of claim 258, wherein the communication is an e-mail communication.

260. The system of claim 251, wherein the communication comprises an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication or a Gopher communication.

261. The system of claim 251, wherein the system processor is further programmed pr adapted to provide an interface for modifying the at least one whitelist.

262. The system of claim 261, wherein the system processor is further programmed or adapted to receive information from the provided interface and apply changes to at least one whitelist based on information received from the interface.

263. The system of claim 261, wherein the interface provides for manual editing of the at least one whitelist.

264. The system of claim 251, wherein the system processor is further programmed or adapted to determine deliverability of a received communication by applying one or more tests.

265. The system of claim 264, wherein received communications determined to be undeliverable are quarantined, discarded, or forwarded.

266. The system of claim 264, wherein the system processor is further programmed or adapted to forward the received communication for delivery if it was determined to be deliverable.

267. The system of claim 264, wherein the system processor applies each of the one or more tests in a parallel fashion.

268. The system of claim 264, wherein the system processor applies each of the one or more tests in a sequential fashion.

269. The system of claim 264, wherein the system data store stores configuration information and wherein the system processor applies each of the one or more tests based upon configuration information stored in the system data store.

270. The system of claim 264, wherein the system processor determines deliverability by calculating an level of trust.

271. The system of claim 270, wherein the system processor determines deliverability by comparing the level of trust to a threshold level.

272. The system of claim 264, wherein the system processor determines whether to deliver a received communication further based upon configuration information stored in the system data store.

273. The system of claim 272, wherein the configuration information comprises communication types, confidence information, time period information, or combinations thereof.

274. The system of claim 264, wherein the system processor is further programmed or adapted to select the one or more tests to determine deliverability.

275. The system of claim 274, wherein the system processor selects the one or more tests based upon communication type, configuration information, or combinations thereof.

276. The system of claim 274, wherein the system processor is further programmed or adapted to compare a received communication to the at least one whitelist and wherein the system processor selects the one or more tests based upon the comparison.

277. The system of claim 264, wherein the system processor is further programmed or adapted to compare a received communication to the at least one whitelist and to selectively bypass the determination of deliverability based upon the comparison.

278. A method for detecting an unsolicited communication transmitted over a communications network, the method comprising the steps of:

a) providing an interface for manually modifying at least one whitelist;

b) receiving an outbound communication of a type selected from the group consisting of an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication and a Gopher communication;

c) storing the received outbound communication; and

d) modifying the at least one whitelist by adding or modifying an entry on the at least one white list based upon a destination of the received outbound communication.

279. The method of claim 278, and further comprising the steps of:

a) receiving an inbound communication;

b) comparing the received inbound communication to the at least one whitelist;

c) selecting a plurality of trust level tests based on a type associated with the received inbound communication, configuration information, the whilelist comparison or combinations thereof;

d) determining deliverability of the received inbound communication by applying the selected plurality of trust level tests;

e) assigning a confidence level to the communication based upon the determined deliverability; and

f) quarantining, discarding, or forwarding the received communication based on the assigned confidence level.

280. A system for detecting an unsolicited communication transmitted over a communications network, the system comprising:

a) means for receiving an electronic communication;

b) direction determination means for determining if the received electronic communication is inbound or outbound;

c) whitelist modification means for updating at least one whitelist based upon a received communication determined to be outbound by the direction determination means by adding an entry or updating an existing entry in the at least one whitelist based upon the received communication; and

d) communication disposition means for disposing of a received communication determined to be inbound by the direction determination means by:

i) comparing the received inbound communication to the at least one whitelist;

ii) selecting a plurality of trust level tests based on a type associated with the received inbound communication, configuration information, the whilelist comparison or combinations thereof;

iii) assigning a confidence level to the communication based upon the determined deliverability; and

iv) quarantining, discarding, or forwarding the received communication based on the assigned confidence level.

281. Computer readable media storing instruction that upon execution by a system processor cause the system processor to automatically generate a whitelist based upon received outbound communication by performing the steps comprising of:

a) providing an interface for manually modifying at least one whitelist;

b) receiving an outbound communication of a type selected from the group consisting of an e-mail communication, an HTTP communication, an FTP communication, a WAIS communication, a telnet communication and a Gopher communication;

c) storing the received outbound communication; and

d) modifying the at least one whitelist by adding or modifying an entry on the at least one white list based upon a destination of the received outbound communication.

282. A method for secure delivery of electronic communications comprising the steps of:

a) receiving an electronic communication from a communication source for delivery to a predetermined recipient;

b) selecting a delivery mechanism from among a plurality of delivery mechanisms based upon a prioritization of the plurality of delivery mechanisms; and

c) attempting to deliver the electronic communication to the predetermined recipient via the selected delivery mechanism.

283. The method of claim 282, and further comprising the step of d) determining the plurality of delivery mechanisms.

284. The method of claim 283, wherein the step of determining the plurality comprises the step of determining the plurality based upon the communication source, the predetermined recipient, a default configuration or combinations thereof.

285. The method of claim 283, and further comprising the step of e) prioritizing the determined plurality of delivery mechanisms.

286. The method of claim 285, wherein the step of prioritizing the determined plurality comprises the step of prioritizing the determined plurality based upon a rating associated with each delivery mechanism in the determined plurality.

287. The method of claim 286, wherein the rating is based upon a criterion selected from the group consisting of delivery efficiency, delivery cost, delivery security and combinations thereof.

288. The method of claim 282, wherein each of the plurality of delivery mechanisms comprises (a) a base mechanism selected from the group consisting of instant messaging, SMTP, HTTP, and FTP and (b) at least one security option and wherein the prioritization of delivery mechanisms is based upon the at least one security option associated with each delivery mechanism.

289. The method of claim 288, wherein the prioritization places delivery mechanisms including S/MIME as a security option first, delivery mechanisms including PGP as a security option but not S/MIME second, and delivery mechanisms including TLS or SSL as a security option but neither PGP nor S/MIME third.

290. The method of claim 282, and further comprising the step of d) retrieving the prioritization of the plurality of delivery mechanism based upon the predetermined recipient.

291. The method of claim 290, wherein each delivery mechanism has an associated rating and wherein the step of retrieving the prioritization comprises the steps of:

1) identifying delivery mechanisms available for delivery of the received communication based upon the predetermined recipient; and
2) prioritizing the identified delivery mechanisms based upon the rating associated with each identified delivery mechanism.

292. The method of claim 291, wherein the rating associated with each delivery mechanism is based upon a criterion selected from the group consisting of delivery efficiency, delivery cost, delivery security and combinations thereof.

293. The method of claim 282, and further comprising the step of d) retrieving the prioritization of the plurality of delivery mechanism based upon the communication source.

294. The method of claim 282, and further comprising the steps of d) providing the communication source with an interface for designating a prioritization of the plurality of delivery mechanisms and e) receiving the prioritization from the provided interface.

295. The method of claim 282, and further comprising the steps of d) providing an administrator with an interface for designating a prioritization of the plurality of delivery mechanisms and e) receiving the prioritization from the provided interface.

296. The method of claim 282, and further comprising the step of d) if delivery fails, attempting to redeliver the electronic communication to the predetermined recipient by the steps comprising of:

1) selecting a further delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization of the plurality of delivery mechanisms; and

2) attempting to deliver the electronic communication to the predetermined recipient via the further delivery mechanism.

297. The method of claim 296, and if the received communication is determined to require secure delivery, further comprising the step of e) repeating step d) until the received communication is successfully delivered or until exhaustion of all available delivery mechanisms in the plurality.

298. The method of claim 282, wherein each of the plurality of delivery mechanisms comprises a base mechanism and at least one security option and wherein each base mechanism is selected from the group consisting of instant messaging, SMTP, HTTP, and FTP.

299. The method of claim 298, wherein each SMTP delivery mechanism in the plurality includes a security option selected from the group consisting of S/MIME, PGP, TLS, SSL and combinations thereof.

300. The method of claim 298, wherein one or more SMTP delivery mechanisms are SMTP notification mechanisms and wherein the security option is HTTP presentment including one or more encryption options selected from the group consisting of S/MIME, HTTP-S, TLS, SSL and combinations thereof.

301. The method of claim 300, wherein a subset of the SMTP notification mechanisms include a further security option associated with delivery of the SMTP notification, wherein the further security option is selected from the group consisting of S/MIME, PGP, TLS, SSL and combinations thereof.

302. The method of claim 300, where the HTTP presentment security option further includes one or more user authentication requirements.

303. The method of claim 298, wherein each HTTP delivery mechanism in the plurality includes a security option selected from the group consisting of S/MIME, HTTP-S, TLS, SSL and combinations thereof.

304. The method of claim 298, wherein each FTP delivery mechanism in the plurality includes a security option selected from the group consisting of TLS and SSL.

305. The method of claim 298, wherein each of the plurality of delivery mechanisms employs a message level encryption technique.

306. The method of claim 298, wherein each of the plurality of delivery mechanisms employs a channel level encryption technique.

307. The method of claim 306, wherein each of the plurality of delivery mechanisms further employs a message level encryption technique.

308. The method of claim 282, and further comprising the step of d) determining whether the received communication requires secure delivery and wherein steps b) and c) occur if the received communication is determined to require secure delivery.

309. The method of claim 308, and if the received communication is determined to require secure delivery, further comprising the steps of e) providing the communication source with an interface for designating a prioritization of the plurality of delivery mechanisms and f) receiving the prioritization from the provided interface.

310. The method of claim 308, wherein determining whether the received communication requires secure delivery comprises the steps of:

i) parsing the received communication for indicia indicating a desire for secure delivery; and

ii) specifying the received communication as requiring secure delivery based upon the parsing.

311. The method of claim 310, wherein the step of specifying the received communication as requiring secure delivery based upon the parsing comprises specifying the received communication as requiring secure delivery if one or more predetermined keywords are parsed from the received communication.

312. The method of claim 310, wherein the parsing step is performed only on a header portion within the received communication and wherein the step of

specifying the received communication as requiring secure delivery based upon the parsing comprises specifying the received communication as requiring secure delivery if one or more predetermined keywords are parsed from the header.

313. The method of claim 310, wherein the step of parsing the received communication for indicia indicating a desire for secure delivery comprises the step of applying one or more filtering rules to the received communication.

314. The method of claim 313, wherein each of the one or more filtering rules is a content filtering rule, an attachment filtering rule, a source filtering rule or a recipient filtering rule.

315. The method of claim 308, wherein determining whether the received communication requires secure delivery comprises the step of specifying the received communication as requiring secure delivery based upon a configuration specified by an administrator.

316. The method of claim 315, wherein the configuration comprises one or more filtering rules, wherein each such rule is of a type selected from the group consisting of a content filtering rule, an attachment filtering rule, a source filtering rule, a communication size filtering rule, a recipient filtering rule and combinations thereof.

317. A system for securely delivering electronic communications, the system comprising:

a) a system data store capable of storing at least one electronic communication and configuration data associated with a plurality of delivery mechanisms;

b) an interface to a communication network that supports the system's communication with one or more client applications;

c) a system processor in communication with the system data store and the interface, wherein the system processor comprises one or more processing elements and wherein the one or more processing elements are programmed or adapted to:

i) receive an electronic communication from a communication source for delivery to a predetermined recipient via the interface;

ii) store the received communication;

iii) determine whether the received communication requires secure delivery based upon the received communication, the predetermined recipient, the communication source, default configuration data or combinations thereof; and

iv) if the received communication is determined to require secure delivery,

1) determine a plurality of delivery mechanisms based upon the communication source, the predetermined recipient, a default configuration or combinations thereof, wherein each of the plurality of delivery mechanisms comprises a base mechanism and at least one security option and wherein each base mechanism is selected from the group consisting of instant messaging, SMTP, HTTP, FTP, and SMTP notification with HTTP presentment and wherein each security options is a channel level encryption, a message level encryption or a combination thereof;

2) prioritize the delivery mechanisms in the plurality in an order corresponding to most secure to least secure concurrent with, or subsequent to, determining the plurality of delivery mechanisms;

3) select a delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization;

4) attempt to deliver the electronic communication to the predetermined recipient via the selected delivery mechanism using the interface or a second interface allowing communication between the system and a second communication network;

5) if delivery fails, attempt to redeliver the electronic communication to the predetermined recipient by at least:

(a) selecting a further delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization of the plurality of delivery mechanisms;

(b) attempting to deliver the electronic communication to the predetermined recipient via the further delivery mechanism; and

(c) repeating (a) and (b) until exhaustion of delivery mechanisms in the plurality.

318. A system for securely delivering electronic communications, the system comprising:

a) interface means for allowing communication between the system and one or more client applications;

b) storage means for providing data storage capacity sufficient to store at least one communication and configuration data associated with a plurality of delivery mechanisms;

c) means for receiving a communication via interface means and storing the communication in the storage means;

d) determining means for determining whether a received communication requires secure delivery based upon the received communication, the predetermined recipient, the communication source, default configuration data or combinations thereof;

e) delivery means for:

i) determining a plurality of delivery mechanisms based upon the communication source, the predetermined recipient, a default configuration or combinations thereof, wherein each of the plurality of delivery mechanisms comprises a base mechanism and at least one security option and wherein each base mechanism is selected from the group consisting of instant messaging, SMTP, HTTP, FTP, and SMTP notification with HTTP presentment and wherein each security options is a channel level encryption, a message level encryption or a combination thereof;

ii) concurrent with, or subsequent to, determining the plurality of delivery mechanisms, prioritizing the delivery mechanisms in the plurality in an order corresponding to most secure to least secure;

iii) selecting a delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization;

iv) attempting to deliver the electronic communication to the predetermined recipient via the selected delivery mechanism; and

v) if delivery fails, attempting to redeliver the electronic communication to the predetermined recipient by the steps comprising of:

1) selecting a further delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization of the plurality of delivery mechanisms;

2) attempting to deliver the electronic communication to the predetermined recipient via the further delivery mechanism; and

3) repeating 1) and 2) until exhaustion of delivery mechanisms in the plurality.

f) wherein the delivery means comprises encryption means for providing channel level, message level or channel and message level encryption.

319. Computer readable media including instructions that upon execution by a system processor comprising one or more processing elements cause the one or more processing elements to securely deliver electronic messages by performing steps comprising of:

a) receiving an electronic communication from a communication source for delivery to a predetermined recipient;

b) determining whether the received communication requires secure delivery based upon the received communication, the predetermined recipient, the communication source, default configuration data or combinations thereof; and

c) if the received communication is determined to require secure delivery,

i) determining a plurality of delivery mechanisms based upon the communication source, the predetermined recipient, a default configuration or combinations thereof, wherein each of the plurality of delivery mechanisms comprises a base mechanism and at least one security option and wherein each base mechanism is selected from the group consisting of

instant messaging, SMTP, HTTP, FTP, and SMTP notification with HTTP presentment and wherein each security options is a channel level encryption, a message level encryption or a combination thereof;

ii) concurrent with, or subsequent to, determining the plurality of delivery mechanisms, prioritizing the delivery mechanisms in the plurality in an order corresponding to most secure to least secure;

iii) selecting a delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization;

iv) attempting to deliver the electronic communication to the predetermined recipient via the selected delivery mechanism; and

v) if delivery fails, attempting to redeliver the electronic communication to the predetermined recipient by the steps comprising of:

1) selecting a further delivery mechanism from among the plurality of delivery mechanisms based upon the prioritization of the plurality of delivery mechanisms; and

2) attempting to deliver the electronic communication to the predetermined recipient via the further delivery mechanism; and

vi) repeating step v) until exhaustion of delivery mechanisms in the plurality.