(12) United States Patent
Porras et al.

(45) Date of Patent: Mar. 23, 2004

(54) NETWORK SURVEILLANCE

(22) Filed: Sep. 25, 2002
(60) Published: Sep. 25, 2002
(72) Assignee: SRI International, Menlo Park, CA (US)

This patent is subject to a terminal disclaimer.

(62) Related U.S. Application Data

(63) Continuation of application No. 09/558,127, filed on Sep. 8, 2000, now Pat. No. 6,484,203, which is a continuation of application No. 09/188,739, filed on Nov. 9, 1998, now Pat. No. 6,731,338.

(51) Int. Cl. 8 G06F 11/30; G06F 12/14
(52) U.S. Cl. 713/200, 201, 709/225
(56) References Cited

U.S. PATENT DOCUMENTS

4,672,699 A 6/1987 Humphrey et al. 371/21
4,713,028 A 5/1988 Tallman 264/250
5,102,704 A 5/1993Res decay 364/251.01
5,440,723 A 6/1995 Hori et al. 379/335
5,529,659 A 7/1996 McKee et al. 706/224
5,577,742 A 9/1996Steak et al. 379/335

FROM SOURCE FIELDS

REF.

INFORMATION FIELDS

FROM SOURCE FIELDS

REF.

INFORMATION FIELDS

FROM SOURCE FIELDS

REF.

INFORMATION FIELDS

OTHER PUBLICATIONS

Debar et al., "A Neural Network Component for an Intrusion Detection System," © 1992 IEEE.

(37) ABSTRACT

A method of network surveillance includes receiving network packets handled by a network entity and building at least one long-term and at least one short-term statistical profile from a measure of the network packets that monitors data transfers, errors, or network connections. A comparison of the statistical profiles is used to determine whether the difference between the statistical profiles indicates suspicious network activity.

93 Claims, 5 Drawing Sheets

SRI 032517
U.S. PATENT DOCUMENTS

5,790,779 A 8/1998 Modig ........................................ 709/224
5,793,196 A 9/1998 de la Salle ........................................ 709/224
5,793,298 A 7/1998 Keyser et al. ........................................ 709/224
5,795,201 A 7/1998 Sides ........................................ 709/224
5,795,591 A 8/1998 Boyle et al. ........................................ 709/224
5,794,457 A 1/1998 Wlcek ........................................ 709/224
5,790,731 A 8/1998 Coventry et al. ........................................ 709/224
5,802,467 A 12/1998 Katellif et al. ........................................ 709/224
6,022,704 A 4/2000 Paul ........................................ 709/224
6,072,460 A 5/2000 Orchard et al. ........................................ 709/224
6,144,561 A 1/2000 de la Salle ........................................ 707/10
6,294,864 B1 5/2002 Snigel ........................................ 709/224 X
6,253,366 B1 9/2001 Gog et al. ........................................ 709/224
6,420,414 B1 7/2001 Olden ........................................ 712/201
6,420,497 B1 7/2001 Joyce ........................................ 712/201
20000037217 A 3/2000 Mullen et al. ........................................ 709/250
20010037279 A 3/2001 Mullen et al. ........................................ 709/250
20020033280 A 3/2002 Peletre et al. ........................................ 714/04
20020035098 A 3/2002 Mullen et al. ........................................ 713/201
20020031879 A 3/2002 Musumato ........................................ 713/200
20020034415 A 7/2002 Copeland, III ........................................ 713/201
20030037136 A 2/2003 Labovitz et al. ........................................ 709/224

OTHER PUBLICATIONS

Dowell, "The Computerwatch Data Reduction Tool," AT&T Bell Laboratories, Whippany, New Jersey.


Vinals, A., Blue Strokes, Sensor Correlation, and Alert Fusion, www.wizs-symposium.org/raids2000/Materials/Ab-


SRI 032519
FIG. 3
FIG. 4

MONITOR NETWORK PACKETS

BUILD STATISTICAL PROFILES FROM MEASURES DERIVED FROM NETWORK PACKETS

DETERMINE IF STATISTICAL PROFILE IS ANOMALOUS (ABNORMAL)

RESPOND

FIG. 5

RECEIVE EVENT RECORD (e.g. DESCRIBING A PACKET)

DESMITUTE FOR ADDITION TO ONE OF A MULTIPLE OF SHORT-TERM STATISTICAL PROFILES

COMPARE ONE OF THE SHORT-TERM PROFILES TO A CORRESPONDING LONG-TERM STATISTICAL PROFILE
NETWORK SURVEILLANCE

This application is a continuation of U.S. application Ser. No. 08/582,137, filed on Sep. 8, 2000 (now U.S. Pat. No. 6,484,203), which is a continuation of U.S. application Ser. No. 09/188,739, filed Nov. 9, 1998 (now U.S. Pat. No. 6,323,336). These both applications are hereby incorporated by reference.

REFERENCE TO GOVERNMENT FUNDING

This invention was made with Government support under Contract Number F30602-96-C-0254 and F30602-96-C-0187 awarded by DARPA and the Air Force Research Laboratory. The Government has certain rights in this invention.

REFERENCE TO APPENDIX

An appendix consisting of 935 pages is included as part of the specification. The appendix includes material subject to copyright protection. The copyright owner does not object to the facsimile reproduction of the appendix, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights.

BACKGROUND

The invention relates to computer networks.

Computer networks offer users ease and efficiency in exchanging information. Networks too often include conglomerates of integrated commercial and custom-made components, interoperating and sharing information at increasing levels of demand and capacity. Such varying networks, though managing a growing list of needs including transportation, commerce, energy management, communications, and defense.

Unfortunately, the very interconnectivity and sophisticated integration of technology that makes networks such valuable assets also make them vulnerable to attack, and make dependence on networks a potential liability. Numerous examples of planned network attacks, such as the Internet worm, have shown how interconnectivity can be used to spread harmful program code. Accidental outages such as the 1980 ARPAnet collapse and the 1990 AT&T collapse illustrate how seemingly localized triggering events can have globally disastrous effects on widely distributed systems. In addition, organized groups have performed malicious and coordinated attacks against various online targets.

SUMMARY

In general, in one aspect, a method of network surveillance includes receiving network packets (e.g., TCP/IP packets) handled by a network entity and building at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. A comparison of at least one long-term and at least one short-term statistical profile is used to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

Embodiments may include one or more of the following features. The measure may monitor data transfers by monitoring network packet data transfer volumes, data transfer errors, and/or monitoring network packet data transfer volume. The measure may monitor network connections by monitoring network connection requests, network connection details, and/or a correlation of network connection requests and network connection details. The measure may monitor errors by monitoring error codes included in a network packet such as privilege error codes and/or error codes indicating a reason a packet was rejected.

The method may also include responding based on the determining whether the difference between a short-term statistical profile and a long-term statistical profile indicates suspicious network activity. A response may include alerting analysis of network packets and/or routing a communication channel. A response may include transmitting and event record to a network monitor, such as hierarchically higher network monitor and a network monitor that receives event records from multiple network monitors.

The network entity may be a gateway, a router, or a proxy server. The network entity may instead be a virtual private network entity (e.g., a VPN).

In general, in another aspect, a method of network surveillance includes monitoring network packets handled by a network entity including building a long-term and multiple short-term statistical profiles of the network packets. A comparison of one of the multiple short-term statistical profiles with the long-term statistical profile is used to determine whether the difference between the short-term statistical profiles and the long-term statistical profile indicates suspicious network activity.

Embodiments may include one or more of the following. The multiple short-term statistical profiles may monitor different anonymous FTP sessions. Building multiple short-term statistical profiles may include distinguishing packets to identify a short-term statistical profile.

In general, in another aspect, a method of network surveillance includes monitoring network packets from a virtual private network entity and statistically analyzing the received packets to determine whether the packets indicate suspicious network activity. The packets may or may not be decrypted before statistical analysis.

Advantages may include one or more of the following. Using long-term and a short-term statistical profiles from measures that monitor data transfers, errors, or network connections protects network components from intruders. Long-term profiles represent "normal" activity, abnormal activity may be detected without requiring an administrator to catalog each possible attack upon a network. Additionally, the ability to distinguish packets to create multiple short-term profiles for comparisons against a long-term profile enables the system to detect abnormal behavior that may be statistically measured if only a single short-term profile was created.

The scheme of communication network monitors also protects networks from more global attacks. For example, an attack made upon one network entity may cause other entities to be attack. Further, a monitor that collects event reports from different monitors may correlate activity to identify attacks causing disturbances in more than one network entity.
Additionally, statistical analysis of packets handled by a virtual private network enables detection of suspicious activity despite virtual private network security techniques. Other features and advantages will become apparent from the following description, including the drawings, and from the claims.

**BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 is a diagram of network monitors deployed in an enterprise.

FIG. 2 is a diagram of a network monitor that monitors an event stream.

FIG. 3 is a diagram of a resource object that configures the network monitor of FIG. 2.

FIG. 4 is a flowchart illustrating network surveillance.

FIG. 5 is a flowchart illustrating multiple short-term statistical profiles for occupation against a single long-term statistical profile.

FIG. 6 is a diagram of a computer platform suitable for deployment of a network monitor.

**DETAILED DESCRIPTION**

Referring to FIG. 1, an enterprise 10 includes different domains 12a-12c. Each domain 12a-12c includes one or more computers offering local and network services that provide an interface for requests internal and external to the domain 12a-12c. Network services include features common to network operating systems such as email, FTP, TCP/IP, remote login, network file systems, faxes, Kerberos, and SNMP. Some domains 12a-12c may share trust relationships with other domains (other peer-to-peer or hierarchical). Alternatively, domains 12a-12c may operate in complete mistrust of all others, providing outgoing connections only or severely restricting incoming connections. Users may be local to a single domain or may possess accounts or multiple domains that allow them to freely establish connections throughout the enterprise 10.

As shown, the enterprise 10 includes dynamically deployed network monitors 16a-16f that analyze and respond to network activity and can interoperate to form an analysis hierarchy. The analysis hierarchy provides a framework for the recognition of more global threats to interdomain connectivity, including coordinated attempts to infiltrate or destroy connectivity across an entire network enterprise 10. The hierarchy includes service monitors 16a-16c, domain monitors 16a-16e, and enterprise monitors 16f.

Service monitors 16a-16c provide local real-time analysis of network packets (e.g., TCP/IP packets) handled by a network entity 14a-14c. Network entities include gateways, routers, firewalls, or proxy servers. A network entity may also be part of a virtual private network. A virtual private network (VPN) is constructed using public wires to connect nodes. For example, a network could use the Internet as the medium for transporting data and use encryption and other security mechanisms to ensure that only authorized users access the network and that the data cannot be intercepted. A monitor 16a-16f can analyze packets both before and after decryption by a node of the virtual private network.

Information gathered by a service monitor 16a-16c can be disseminated to other monitors 16a-16f, for example, via a subscription-based communication scheme. In a subscription-based scheme, client monitors subscribe to receive analysis reports produced by server monitors. As a monitor 16a-16f produces analysis reports, the monitor 16a-16f disseminates these reports asynchronously to subscribers. Through subscription, monitors 16a-16f distributed throughout a large network are able to efficiently disseminate reports of malicious activity without requiring the overhead of synchronous polling.

Domain monitors 16d-16e perform surveillance over all or part of a domain 12a-12c. Domain monitors 16d-16e correlate intrusion reports disseminated by individual service monitors 16a-16c, providing a domain-wide perspective of activity (or patterns of activity). In addition to domain surveillance, domain monitors 16d-16e can reconfigure system parameters, interfaces with other monitors beyond a domain, and report threats against a domain 12a-12c to administrators. Domain monitors 16d-16e can subscribe to service monitors 16a-16c. Where monitors track among domains 12a-12c exist, domain monitors 16d-16e can establish peer relationships with one another. Peer-to-peer subscription allows domain monitors 16d-16e to share analysis reports produced in other domains 12a-12c. Domain monitors 16d-16e may use such reports to dynamically sensitize their local service monitors 16a-16c to malicious activity that was occurring outside a domain 12a-12c. Domain monitors 16d-16e may also operate within an enterprise hierarchy where they disseminate analysis reports to enterprise monitors 16f for global correlation.

Enterprise monitors 16f correlate analysis activity reports produced across the set of monitored domain 12a-12c. Enterprise 10 surveillance may be used where domains 12a-12c are interconnected under the control of a single organization, such as a large privately owned WAN (Wide Area Network). The enterprise 10, however, need not be stable in its configuration or centrally administered. For example, the enterprise 10 may exist as an emergent entity through new interconnections of domains 12a-12c. Enterprise 10 surveillance is very similar to domains 12a-12c surveillance: an enterprise monitor 16f subscribes to various domain monitors 16d-16e, just as the domain monitors 16d-16e subscribe to various service monitors 16a-16c. The enterprise monitor 16f (or monitors, as it would be important to avoid centralizing any analysis) focuses on network-wide threats such as Internet worm-like attacks, attacks repeated against common network services across domains, or coordinated attacks from multiple domains against a single domain. As an enterprise monitor 16f recognizes commonalities in intrusion reports across domains (e.g., the spreading of a worm or a mail system attack repeated throughout the enterprise 10), the monitor 16f can help domains 12a-12c counter the attack and can sensitize other domains 12a-12c to such attacks before they are affected. Through correlation and sharing of analysis reports, reports of problems found by one monitor 16a-16f may propagate to other monitors 16a-16f throughout the network. Interdomain event analysis is vital to addressing more global, information attacks against the entire enterprise 10.
that enhances encapsulation of monitor functions and eases integration of third-party intrusion-detection tools 28, 30.

Each monitor 16 can analyze event records that form an event stream. The event stream may be derived from a variety of sources such as TCP/IP network packet contents or event records containing analysis reports disseminated by other monitors. For example, an event record can be formed from data included in the header and data segment of a network packet. The volume of packets transmitted and received, however, dictates careful assessment of ways to select and organize network packet information into event record streams.

Selection of packets can be based on different criteria. Streams of event records can be derived from discarded traffic (i.e., packets not allowed through the gateway because they violate filtering rules), pass-through traffic (i.e., packets allowed into the internal network from external sources), packets having a common protocol (e.g., all ICMP (Internet Control Message Protocol) packets that reach the gateway), packets involving network connection management (e.g., SYN, RESERVE, ACK, [window receive]), and packets targeting ports to which an administrator has not assigned any network service and that also remain unblocked by the firewall. Event streams may also be based on packet source addresses (e.g., packets whose source addresses match well-known external sites such as satellite offices or have raised suspicion from other monitoring efforts) or destination addresses (e.g., packets whose destination addresses match a given internal host or workstation). Selection can also implement application-layer monitoring (e.g., packets targeting a particular network service or application). Event records can also be produced from other sources of network packet information such as report logs produced by network entities. Event streams can be of very fine granularity. For example, a different stream might be derived for commands received from different commercial web-browsers since each web-browser produces different characteristic network activity.

A monitor 16 can also construct interval summary event records, which contain accumulated network traffic statistics (e.g., number of packets and number of kilobytes transferred). These event records are constructed at the end of each interval (e.g., once per N seconds). Event records are forwarded to the analysis engines 22, 24 for analysis.

The profile engines 22 can use a wide range of multidimensional statistical measures to profile network activity indicated by an event stream. A statistical score represents how closely currently observed usage corresponds to the established patterns of usage. The profile engines 22 separate profile management and the mathematical algorithms used to access the anomaly of events. The profile engines 22 may use a statistical analysis technique described in A. Valdes and D. Anderson, "Statistical Methods for Computer Usage Anomaly Detection Using NIDES", Proceedings of the Third International Workshop on Rough Sets and Soft Computing, January 1995, which is incorporated by reference in its entirety. Such an engine 22 can profile network activity via one or more variables called measures. Measures can be categorized into four classes: categorical, continuous, intensity, and event distribution measures.

Categorical measures assume values from a discrete, unordered set of possibilities. Examples of categorical measures include network source and destination address, commands (e.g., commands that control data transfer and manage network connections), protocols, error codes (e.g., privilege violations, malformed service requests), and formatted packet codes, and port identifiers. The profiler engine 22 can build empirical distributions of the category values encountered, even if the list of possible values is open-ended. The engine 22 can have mechanisms for "aging out" categories whose long-term probabilities drop below a threshold.

Continuous measures assume values from a continuous or ordinal set. Examples include inter-event times (e.g., difference in time stamps between consecutive events from the same stream), counting measures such as the number of errors of a particular type observed in the recent past, the volume of data transfers over a period of time, and network traffic measures (number of packets and number of kilobytes). The profiler engine 22 can store continuous measures by binning values appropriate to the range of values of the underlying measure, and then tracking the frequency of observation of each value range. In this way, multi-modal distributions are accommodated and the computationally costly use of function approximations for categorical measures is avoided. Continuous measures are useful not only for intrusion detection, but also for assessing the health and status of the network from the perspective of connectivity and throughput. For example, a measure of traffic volume maintained can detect an abnormal loss in the data rate of received packets when this volume falls outside historical norms. This sudden drop can be specific both to the network entity being monitored and to the time of day (e.g., the average sustained traffic rate for a major network activity is much different at 11:00 a.m. than at midnight).

Intensity measures reflect the intensity of the event stream (e.g., number of ICMP packets) over specified time intervals (e.g., 1 minute, 15 minutes, and 1 hour). Intensity measures are particularly suited for detecting flooding attacks, while also providing insight into other anomalies.

Event distribution measures are meta-measures that describes how other measures in the profile are affected by each event. For example, an "ls" command in an FTP session affects the directory measure, but does not affect measures related to file transfer. This measure is not interesting for all event streams. For example, all network-traffic event records affect the same measures (number of packets and kilobytes) defined for that event stream, so the event distribution does not change. On the other hand, event distribution measures are useful in correlating analysis performed by a monitor 16a-16f that receives reports from other monitors 16a-16f.

The system maintains and updates a description of behavior with respect to these measure types in an updated profile. The profile is subdivided into short-term and long-term profiles. The short-term profile accumulates values between updates, and exponentially ages (e.g., weights data based on how long ago the data was collected) values for comparison to the long-term profile. As a consequence of the aging mechanism, the short-term profile characteristics reflect activity, whereas "recent" is determined by a dynamically configurable aging parameter. At update time (typically, a unit of few system activity), the update function folds the short-term values observed since the last update into the long-term profile, and the short-term profile is cleared. The long-term profile is itself slowly aged to adapt to changes in subject activity. Anomaly scoring compares updated surrogates in the short-term profile against the long-term profile. As all evaluations are done against empirical distributions, no assumptions are made, and multi-modal and categorical distributions are accommodated. Furthermore, the algorithms require no a priori knowledge of intrusive or exceptional activity.

SRI 032527
The statistical algorithm adjusts a short-term profile for the measure values observed in the event record. The distribution of recently observed values is compared against the long-term profile, and a distance between the two is obtained. The difference is compared to a historically adaptive deviation. The empirical distribution of this deviation is transformed to obtain a score for the event. Anomalies are those whose scores exceed a historically adaptive score threshold based on the empirical score distribution. This comparative approach handles all measure types and makes no assumptions on the modality of the distribution for continuous measures.

Profiles are provided to the computational engine as classes defined in the resource object 32. The mathematical functions for anomaly scoring, profile maintenance, and updating do not require knowledge of the data being analyzed beyond what is encoded in the profile class. Event collection interoperability supports translation of the event streams to the profile and measure classes. As that point, analysis for different types of monitored entities is mathematically similar. This approach impacts great flexibility to the analysis in that scaling memory occupancy, update frequency, measure type, and so on are tailored to the network entity being monitored.

The measure types described above can be used individually or in combination to detect network packet attributes characteristic of intrusion. Such characteristics include large data transfers (e.g., moving or downloading files), an increase in errors (e.g., an increase in privilege violations or network packet rejections), network connection activity, and abnormal changes in network volume.

As shown, the monitor 14 also includes a signature engine 24. The signature engine 24 maps an event stream against abstract representations of events that are known to indicate undesirable activity. Signature-analysis objectives depend on which layer in the hierarchical analysis scheme the signature engine operates. Service intrusion 16A-16C signature engines 24 attempt to monitor for attempts to penetrate or interfere with the domain's operation. The signature engine scans the event stream for events that represent the detection of known attacks against the service, or other activity that stands alone as warranting a response from the monitor. Above the service layer, signature engines 24 scan the aggregate of intrusion reports from service monitors in an attempt to detect more global coordinated attack scenarios or scenarios that exploit interdependencies among network services. Layering signature-engine analysis enables the engines 24 to avoid ill-guided searches among incorrect signature paths in addition to distributing the signature analysis.

A signature engine 24 can detect, for example, address spoofing, tunneling, source routing, SATAN attacks, and abuse of ICMP messages ("RedEye" and "Destination Unreachable" messages in particular). Threshold analysis is a rudimentary, inexpensive signature-analysis technique that records the occurrence of specific events and, as the name implies, detects when the number of occurrences of that event surpasses a reasonable count. For example, monitors can execute thresholds to monitor activity such as the number of fingers, pings, or failed login requests to accounts such as guest, demo, visitor, anonymous FTP, or employees who have departed the company.

Signature engine 24 can also examine the data portion of packets in search of a variety of transactions that indicate suspicious, if not malicious, intentions by an external client. The signature engine 24, for example, can parse FTP traffic traveling through the firewall or router for unwanted transfers of configuration or specific system data, or anonymous requests to access non-public portions of the directory structure. Similarly, a monitor can analyze anonymous FTP sessions to ensure that the file retrievals and uploads/modifications are limited to specific directories. Additionally, signature analysis capability can extend to anomaly analysis of complex and deep packet, but highly useful, services like HTTP or Gopher.

Signature analysis can also scan traffic directed at assigned ports (i.e., ports to which the administrator has not assigned a network service). Here, packet scanning can be used to study network traffic after some threshold volume of traffic, directed at an assigned port, has been exceeded. A signature engine 24 can also employ a knowledge base of known well-known network-service protocol traffic (e.g., FTP, Telnet, SMTP, HTTP). The signature engine 24 then determines whether the unknown port traffic matches any known packet sets. Such comparisons could lead to the discovery of network services that have been installed without an administrator's knowledge.

The analysis engines 22, 24 receive large volumes of events and produce smaller volumes of intrusion or suspicious reports that are then fed to the resolver 20. The resolver 20 is an expert system that receives the intrusion and suspicious reports produced by the analysis engines 22, 24 and reports produced externally by other analysis engines to which it subscribes. Based on these reports, the resolver 20 invokes responses. Because the volume of intrusion and suspicious reports is lower than the volume of events received by the analysis engines 22, 24, the resolver 20 can afford the more sophisticated demands of configuration maintenance and managing the response handling and external interfaces necessary for monitor operation. Furthermore, the resolver 20 adds to extensibility by providing the subscription interface through which third-party analysis tools 28, 30 can interact and participate in the hierarchical analysis scheme.

Upon its initialization, the resolver 20 initiates subscription and subscription sessions with those monitors 16A-16F whose identities appear in the monitor's 16 subscription list (Fig. 15). The resolver 20 also handles all incoming requests by subscribers, which must authenticate themselves to the resolver 20. Once a subscription session is established with a subscriber monitor, the resolver 20 acts as the primary interface through which configuration requests are received and intrusive reports are disseminated.

Thus, resolvers 20 can request and receive reports from other resolvers at lower layers in the analysis hierarchy. The resolver 20 forwards analysis reports received from subscribers to the analysis engines 22, 24. This tiered collection and correlation of analysis results allows monitors 16A-16F to represent and profile global malicious or anomalous activity that is not visible locally.

In addition to external-interface responsibilities, the resolver 20 operates as a fully functional decision engine, capable of invoking real-time response measures to respond to malicious or anomalous activity reports produced by the analysis engines. The resolver 20 also operates as the center of intranetwork communication. As the analysis engines 22, 24 build intrusion and suspicious reports, they propagate these reports to the resolver 20 for further correlation, response, and dissemination to other monitors 16A-16F. The resolver 20 can also submit runtime configuration requests to the analysis engines 22, 24, for example, to increase or...
provide the resolver 28 with intrusion or suspicious reports either asynchronously or upon request. Similarly, the analysis engines 22, 24 are responsible for establishing and maintaining a communication link with an event collection method (or event filter) and prompting the reconfiguration of the collection method’s filtering sensibility when necessary.

Intermediate communication also operates using the subscription-based hierarchy. A domain monitor 16b-16c subscribes to the analysis results produced by service monitors 16b-16c, and these propagates its own analytical reports to its parent enterprise monitor 16f. The enterprise monitor 16f operates as a client to one or more monitor domains 16d-16e, allowing them to correlate and model enterprise-wide activity from the domain-layer results. Domain monitors 16d-16e operate as servers to the enterprise monitor 16f, and as clients to the service monitors 16b-16c deployed throughout their domain 16a-16c. This message scheme can operate substantially the same if correlation and monitoring are demanded at higher layers of abstraction beyond enterprise 10 analysis.

Intramonitor and intermonitor programming interfaces are substantially the same. These interfaces can be divided into five categories of interoperation: channel initialization and translation, channel systemization, dynamic configuration, server probing, and report/event dissemination. Clients are responsible for initiating and terminating channel sessions with servers. Clients are also responsible for assigning channel systemization in the event of errors in message sequencing or periods of failed or slow response (i.e., “I’m alive” confirmations). Clients may also submit dynamic configuration requests to servers. For example, an analysis engine 22, 24 may request an event collection method to modify its filtering semantics. Clients may also probe servers for report summaries or additional event information. Lastly, servers may send clients intrusion/suspicious reports in response to client probes or in an asynchronous dissemination mode.

The second part of the message system framework involves specification of a transport mechanism used to establish a given communication channel between monitors 16b-16f or possibly between a monitor 16b-16f and a third-party security module. All implementation dependencies within the message system framework are addressed by plugin transport modules. Transport modules are specified to the participating intrusion-detection modules, their respective hosts, and potentially to the network—should the modules require cross-platform interoperation. Establishing a monitor 16a-16f may involve incorporation of the necessary transport module(s) (for both internal and external communication).

Transport modules that handle intramonitor communication may be different from the transport modules that handle intraserver communication. This allows the intramonitor transport modules to address security and reliability issues differently than how the intraserver transport modules address security and reliability. While intracommunication may more commonly involve interprocess communication within a single host, intraserver communication will more commonly involve cross-platform networked interoperation. For example, the intramonitor transport mechanisms may employ unnamed pipes which provides a kernel-enforced private interprocess communication channel between the monitor 16f. This assumes a process hierarchy within the monitor 16f architecture. The monitor 16f internal transport system will more likely export data through unrelated network connections and thus require more extensive security management. To ensure the security and integrity of the message

SRI 032529
exchange, the external transport may support public/private key authentication protocols and session key exchange. Using this same interface, third-party analysis tools may authenticate and exchange analysis results and configuration information in a well-defined, secure manner.

The pluggable transport permits flexibility in negotiating security features and protected usage with third parties. Incorporation of a commercially available network management system can deliver monitoring results relating to security, reliability, availability, performance, and other attributes. The network management system may in turn subscribe to monitor produced results in order to influence network reconfiguration.

All monitors (service, domain, and enterprise) 16a-16f use the same monitor code-base. However, monitors may include different resource objects 32 having different configuration data and methods. This reusable software architecture can reduce implementation and maintenance efforts. Customizing and dynamically configuring a monitor 16 thus becomes a question of building and/or modifying the resource object 32.

Referring to FIG. 3, the resource object 32 contains the operating parameters for each of the monitor's 16 components as well as the analysis semantics (e.g., the profiler engine's 22 measure and category definitions, or the signature engine's 24 penetration rule-base) necessary to process an event stream. After defining a resource object 32 to implement a particular set of analyses on an event stream, the resource object 32 may be reused by other monitors 16 deployed to analyze equivalent event streams. For example, the resource object 32 for a domain's router may be reused as other monitors 16 are deployed for other routers in a domain. A library of resource objects 32 provides prefabricated resource objects 32 for commonly available network utilities.

The resource object 32 provides a pluggable configuration module for tuning the generic monitor code-base to a specific event stream. The resource object 32 includes configurable event structures 34, analysis unit configuration 36a-36n, engine configuration 40a-40n, resource configuration 42, decision unit configuration 44, subscription list data 46, and response methods 48.

Configurable event structures 34 define the structure of event records and analyze analysis records. The monitor code-base maintains no internal dependence on the format or form of any given event stream or the analysis results produced from analyzing the event streams. Rather, the resource object 32 provides a universally applicable syntax for specifying the structure of event records and analysis results. Event records are defined based on the contents of an event stream(s). Analysis result structures are used to package the findings produced by analysis engines. Event records and analysis results are defined similarly to allow the event hierarchically analyzing results as event records by subscriber monitors.

Event-collection methods 36 gather and parse event records for analysis engine processing. Processing by analysis engines is controlled by engine configuration 40a-40n variables and data structures that specify the operating configuration of a finished monitor's analysis engine(s). The resource object 32 maintains a separate collection of operating parameters for each analysis engine instantiated in the monitor 16. Analysis unit configuration 36a-36n include configuration variables that define the semantics employed by the analysis engines to process the event stream.

The resource configuration 42 includes operating parameters that specify the configuration of the resource's internal modules. The decision unit configuration 44 describes semantics used by the resource's decision unit for processing the analysis results from the various analysis engines. The semantics include the response criteria used to invoke countermeasures based on the resource object 32. The resource object 32 may also include response methods 48. Response methods 48 include preprogrammed countermeasures that the monitor may invoke as event records are received. A response method 48 includes evaluation metrics for determining the circumstances under which the method should be invoked. These metrics include a threshold metric that corresponds to the measure values and scores produced by the profiler engine 22 and security metrics that correspond to subsets of the associated attack sequences defined within the resource object 32.

Countermeasures range from very passive responses, such as report dissemination to other monitors 16a-16f or administrators, to highly aggressive actions, such as sending a communication channel or the reconfiguring of logging facilities within network components (e.g., routers, firewalls, network services, audit daemons). An active response may involve午餐 and validate the profiles of network services or other assets to ensure that privileged network services have not been subverted. Monitors 16a-16f may invoke probes to an extent to gather as much counterintelligence about the source of suspicious traffic by using features such as traceroutes or finger.

The resource object 32 may include a subscription list 46 that includes information necessary for establishing subscription-based communication sessions, which may include network address information and public keys used by the monitor to authenticate potential clients and servers. The subscription list 46 enables transmission or reception of messages that report malicious or anomalous activity between monitors. The most obvious examples where relationships are important involve interdependencies among network services that make local policy decisions. For example, the interdependencies between access checks performed during network file system mounting and the IP mapping of the DNS service. An unexpected event monitored by the network file system service may be responded to differently if the DNS monitor informs the network file system monitor of suspicious updates to the mount requestor's DNS mapping.

The contents of the resource object 32 are defined and utilized during monitor 16 initialization. In addition, these fields may be modified by internal monitor 16 components, and by authorized external clients using the monitor's 16 API. Modifying the resource object 32 permits adaptive analysis of an event stream, however, it also introduces a potential stability problem if dynamic modifications are not tightly restricted to avoid cyclic modifications. To address this issue, monitors 16 can be configured to accept configuration requests from only higher-level monitors 16.

Referring to FIG. 4, a monitor performs network surveillance by monitoring 66 a stream of network packets. The monitor builds a statistical model of network activity from the network packets, for example, by building 68 a long-term and short-term statistical profile from network packets from the network packets. The measures include measures that can show malicious network activity characteristic of network intrusions such as measures that describe data transfers, network connections, privilege and network errors, and abnormal levels of network traffic. The monitor can compare 70 the long-term and short-term statistical profiles to detect suspicious network activity. Based on this comparison, the monitor can respond 72 by reporting the
activity to another monitor or by executing a command-line response. More information can be found in P. Porras and A. Valdes "Live Traffic Analysis of TCP/IP Gateways", Network and Distributed Systems Security Symposium, March 1998, which is incorporated by reference in its entirety.

A few examples can illustrate this method of network surveillance. Network intrusion frequently causes large data transfers, for example, when an intruder seeks to download sensitive files or replace system files with harmful substitu- tutes. A statistical profile to detect anomalies data transfers might include a continuous measure of file transfer size, a categorical measure of the source or destination directory of the data transfer, and an intensity measure of commands corresponding to data transfers (e.g., commands that download data). These measures can detect a wide variety of data transfer techniques such as a large volume of small data transfers, sequential downloading large files on mass. The monitor may distinguish between network packets based on the time such packets were received by the network entities, permitting statistical analysis to distinguish between normal data transfer during a weekday and an abnormal data transfer on a weekend evening.

Attempted network intrusion may also produce suspicious levels of errors. For example, categorical and intensity measures derived from privilege errors may indicate attempts to access protected files, directories, or other network assets. Of course, privilege errors occur during normal network operation as users mistype commands or attempt to perform an operation unknowingly prohibited. By comparing the long-term and short-term statistical profiles, a monitor can distinguish between normal error levels and levels indicative of intrusion without broadcasting a network administrator in the task of arbitrarily setting an unvarying threshold. Other measures based on errors, such as codes describing why a network entity rejected a network packet, enable a monitor to detect attempts to initiate a network with suspicious packets.

Attempted network intrusion can also be detected by measures derived from network connection information. For example, a measure may be formed from the correlation (e.g., a ratio or a difference) of the number of SYN connection request messages with the number of SYN-ACK connection acknowledgment messages and/or the number of ICMP messages sent. Generally, SYN requests received should balance with respect to the total of SYN-ACK and ICMP messages sent. That is, flow into and out of a network entity should be conserved. An imbalance can indicate repeated unsuccessful attempts to connect with a system, perhaps corresponding to a methodical search for an entry point to a system. Alternatively, intensity measures of transport-layer connection requests, such as a volume analysis of SYN-RST messages, could indicate the occurrence of a SYN attack against port availability or possibly port-scanning. Variants of this can include intensity measures of TCP/FIN messages, considered a more stealthy form of port scanning.

Many other measures can detect network intrusion. For example, "doorbell ringing," testing a variety of potentially valid commands to gain access (e.g., trying to access a "superuser" or "root") and a password of "superuser"

... the monitor may also "disinfect.

That is, the monitor may create and update CA or CAs more than one short-term profile for comparison against a single long-term profile by identifying one of the multiple short-term profiles that will be updated by an event record in an event stream. For example, at any time a network entity may handle several FTP "anonymous" sessions. If each network packet for all anonymous sessions were placed in a single short-term statistical profile, potentially intrusive activity of one anonymous session may be statistically overwhelmed by non-intrusive sessions. By creating and updating short-term statistical profiles for each anonymous session, each anonymous session can be compared against the long-term profile of a normal FTP anonymous session. Detrimental can be done for a variety of sessions including HTTP sessions (e.g., a short-term profile for each browser session).

Referring to FIG. 5, a computer platform 14 suitable for executing a network monitor 16 includes a display 56, a keyboard 54, a pointing device 58 such as a mouse, and a digital computer 56. The digital computer 56 includes memory 62, a processor 60, a large storage device 64, and other customary components such as a memory bus and peripheral bus. The platform 14 may further include a network interface 52.

Mass storage device 64a can store instructions that form the monitor 16. The instructions may be transferred to memory 62 and processor 60 in the course of operation. The instructions 14 can cause the display 56 to display an interface such as a graphical user interface. Of course, instructions may be stored on a variety of mass storage devices such as a floppy disk 64b, CD-ROM 64c, or PROM (not shown).

Other embodiments are within the scope of the following claims.
What is claimed is:

1. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:
   deploying a plurality of network monitors in the enterprise network;
   detecting, by the network monitors, suspicious network activity based on analysis of network traffic data selected from one or more of the following categories: network packet data transfer errors, network packet data volume, network connection requests, network connection duration, error codes included in a network packet, network connection acknowledgements, and network packets indicative of well-known network-service protocols;
   generating, by the monitors, reports of said suspicious activity and automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors;

2. The method of claim 1, wherein integrating comprises correlating intrusion reports reflecting underlying commonalities.

3. The method of claim 1, wherein integrating further comprises invoking countermeasures to a suspected attack.

4. The method of claim 1, wherein the plurality of network monitors includes an API for encapsulation of monitor functions and integration of third-party tools.

5. The method of claim 1, wherein the enterprise network is a TCP/IP network.

6. The method of claim 1, wherein the network monitors are deployed at one or more of the following facilities of the enterprise network: gateways, routers, proxy servers.

7. The method of claim 1, wherein at least one of said network monitors utilizes a statistical detection method.

8. The method of claim 1, wherein deploying the network monitors includes deploying a plurality of service monitors among multiple domains of the enterprise network.

9. The method of claim 8, wherein receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

10. The method of claim 1, wherein deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

11. The method of claim 10, wherein receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

12. The method of claim 10, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

13. An enterprise network monitoring system comprising:
   a plurality of network monitors deployed within an enterprise network, said plurality of network monitors detecting suspicious network activity based on analysis of network traffic data selected from one or more of the following categories: network packet data transfer errors, network packet data volume, network connection requests, network connection duration, error codes included in a network packet, network connection acknowledgements, and network packets indicative of well-known network-service protocols;
   said network monitors generating reports of said suspicious activity; and
   one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

14. The system of claim 13, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

15. The system of claim 13, wherein the integration further comprises invoking countermeasures to a suspected attack.

16. The system of claim 13, wherein the plurality of network monitors includes an application programming interface (API) for encapsulation of monitor functions and integration of third-party tools.

17. The system of claim 13, wherein the enterprise network is a TCP/IP network.

18. The system of claim 13, wherein the network monitors are deployed at one or more of the following facilities of the enterprise network: gateways, routers, proxy servers.

19. The system of claim 13, wherein the plurality of network monitors includes a plurality of service monitors among multiple domains of the enterprise network.

20. The system of claim 13, wherein the domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

21. The system of claim 13, wherein the plurality of network monitors include a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

22. The system of claim 21, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

23. The system of claim 21, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

24. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:
   deploying a plurality of network monitors in the enterprise network, wherein the enterprise network is a virtual private network (VPN);
   detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
   generating, by the monitors, reports of said suspicious activity; and
   automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

25. The method of claim 24, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

26. The method of claim 24, wherein said integrating further comprises invoking countermeasures to a suspected attack.

27. The method of claim 24, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

28. The method of claim 24, wherein said network traffic data is selected from one or more of the following categories: network packet data transfer commands, network
packet data transfer errors, network packet data volume, network connection requests, network connection details, error codes included in a network packet.

29. The method of claim 24, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

30. The method of claim 29, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor’s associated network domain.

31. The method of claim 24, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

32. The method of claim 31, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

33. The method of claim 31, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

34. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a gateway;

detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

35. The method of claim 34, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

36. The method of claim 34, wherein said integrating further comprises invoking countermeasures to a suspected attack.

37. The method of claim 34, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

38. The method of claim 34, wherein said network traffic data is selected from one or more of the following categories: network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection details, error codes included in a network packet.

39. The method of claim 34, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

40. The method of claim 39, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor’s associated network domain.

41. The method of claim 34, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

42. The method of claim 41, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

43. The method of claim 41, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

44. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a gateway;
detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

45. The method of claim 44, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

46. The method of claim 44, wherein said integrating further comprises invoking countermeasures to a suspected attack.

47. The method of claim 44, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

48. The method of claim 44, wherein said network traffic data is selected from one or more of the following categories: network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection details, error codes included in a network packet.

49. The method of claim 44, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

50. The method of claim 49, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor’s associated network domain.

51. The method of claim 44, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

52. The method of claim 51, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

53. The method of claim 51, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

54. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a proxy server;
detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

55. The method of claim 54, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

56. The method of claim 54, wherein said integrating further comprises invoking countermeasures to a suspected attack.
57. The method of claim 54, wherein said deploying the network monitors includes an API for encapsulation of monitor functions and integration of third-party tools.

58. The method of claim 54, wherein said network traffic data is selected from one or more of the following categories: (network packet data transfer commands, network packet data transfer errors, network packet data volumes, network connection requests, network connection details, error codes included in a network packet).

59. The method of claim 59, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

60. The method of claim 59, wherein said deploying and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

61. The method of claim 54, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

62. The method of claim 61, wherein said deploying and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

63. The method of claim 61, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

64. A computer-readable method of hierarchical event monitoring and analysis within an enterprise network comprising:

- deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a firewall;
- detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
- generating, by the network monitors, reports of said suspicious activity;
- and automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

65. The method of claim 64, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

66. The method of claim 64, wherein said integrating further comprises invoking countermeasures to a suspected attack.

67. The method of claim 64, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

68. The method of claim 64, wherein said network traffic data is selected from one or more of the following categories: (network packet data transfer commands, network packet data transfer errors, network packet data volumes, network connection requests, network connection details, error codes included in a network packet).

69. The method of claim 64, wherein said deploying the network monitors includes placing a plurality of service monitors within the domain monitor's associated network domain.

70. The method of claim 69, wherein said deploying and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

71. The method of claim 64, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

72. The method of claim 71, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

73. The method of claim 71, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

74. An enterprise network monitoring system comprising:

- a plurality of network monitors deployed within an enterprise network, wherein the enterprise network is a virtual private network (VPN), wherein said network monitors detecting suspicious network activity based on analysis of network traffic data;
- said network monitors generating reports of said suspicious activity; and
- one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

75. The system of claim 74, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

76. The system of claim 74, wherein the integration further comprises invoking countermeasures to a suspected attack.

77. The system of claim 74, wherein the plurality of network monitors include an application programming interface (API) for encapsulation of monitor functions and integration of third-party tools.

78. The system of claim 74, wherein said network traffic data is selected from one or more of the following categories: (network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection details, error codes included in a network packet).

79. The system of claim 74, wherein the plurality of network monitors includes a plurality of service monitors among multiple domains of the enterprise network.

80. The system of claim 79, wherein a domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

81. The system of claim 74, wherein the plurality of network monitors includes a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

82. The system of claim 81, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

83. The system of claim 81, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

84. An enterprise network monitoring system comprising:

- a plurality of network monitors deployed within an enterprise network, wherein at least one of the network monitors is deployed at one or more of the following facilities of the enterprise network: (gateways, routers, proxy servers, firewalls), said plurality of network monitors detecting suspicious network activity based on analysis of network traffic data;
- said network monitors generating reports of said suspicious activity; and
- one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

SRI 032534
83. The system of claim 84, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

84. The system of claim 84, wherein the integration further comprises invoking countermeasures to a suspected attack.

87. The system of claim 84, wherein the plurality of network monitors include an application programming interface (API) for encapsulation of monitor functionalities and integration of third-party tools.

88. The system of claim 84, wherein said network traffic data is selected from one or more of the following categories: network packet data, transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet.

89. The system of claim 84, wherein the plurality of network monitors includes a plurality of service monikers among multiple domains of the enterprise network.

90. The system of claim 89, wherein a domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

91. The system of claim 84, wherein the plurality of network monitors include a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

92. The system of claim 91, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

93. The system of claim 91, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

SRI 032535