US 20040143753A1

(19) United States

(12) Patent Application Publication

Hernacki et al.

(10) Pub. No.: US 2004/0143753 A1

(43) Pub. Date: Jul. 22, 2004

(54) NETWORK RISK ANALYSIS

(75) Inventors: **Brian Hernacki**, San Carlos, CA (US); **Jeremy Bennett**, Mountain View, CA (US)

Correspondence Address:
VAN PELT & YI LLP
10050 N. FOOTHILL BLVD #200
CUPERTINO, CA 95014 (US)

(73) Assignee: **Symantec Corporation**

(21) Appl. No.: **10/349,155**

(22) Filed: **Jan. 21, 2003**

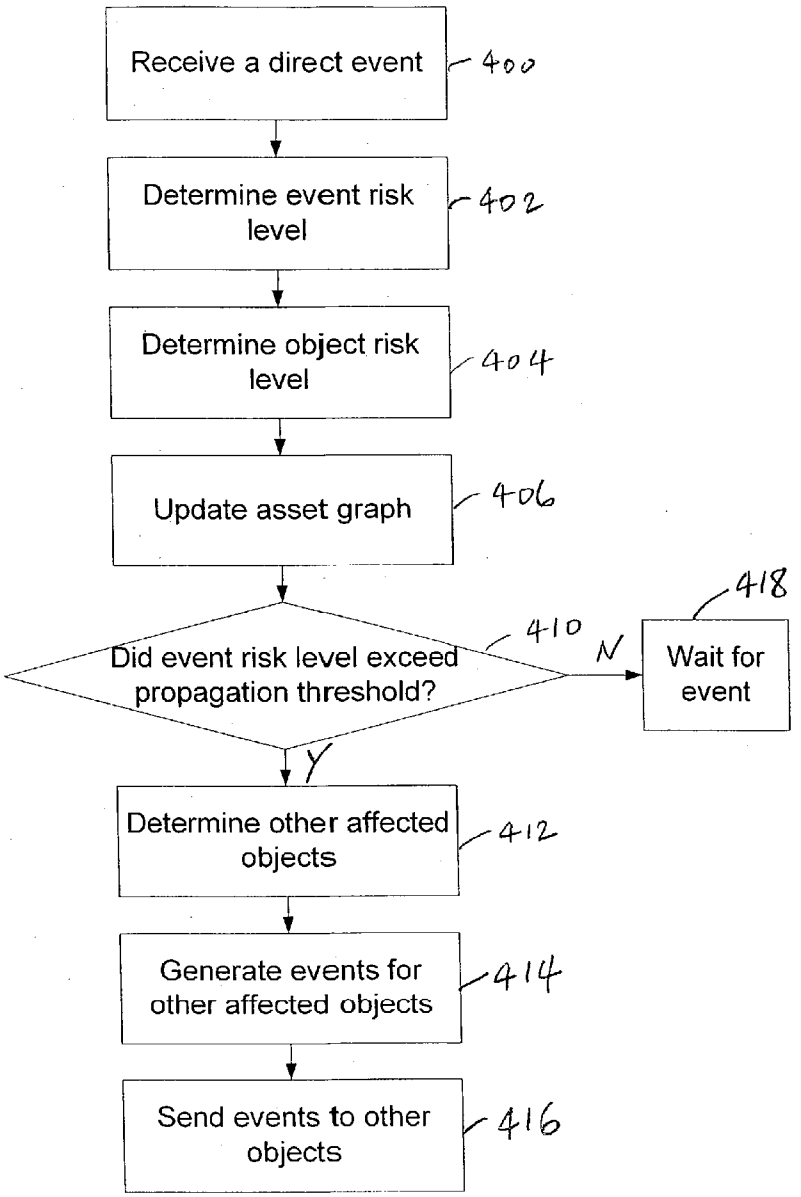
Publication Classification

(51) **Int. Cl.**[7] **H04L 9/00**

(52) **U.S. Cl.** **713/200**

(57) **ABSTRACT**

A system and method are disclosed for analyzing security risks in a computer network. The system constructs asset relationships among a plurality of objects in the computer network and receives an event associated with a selected object, where the event has an event risk level. The system also propagates the event to objects related to the selected object if the event risk level exceeds a propagation threshold.



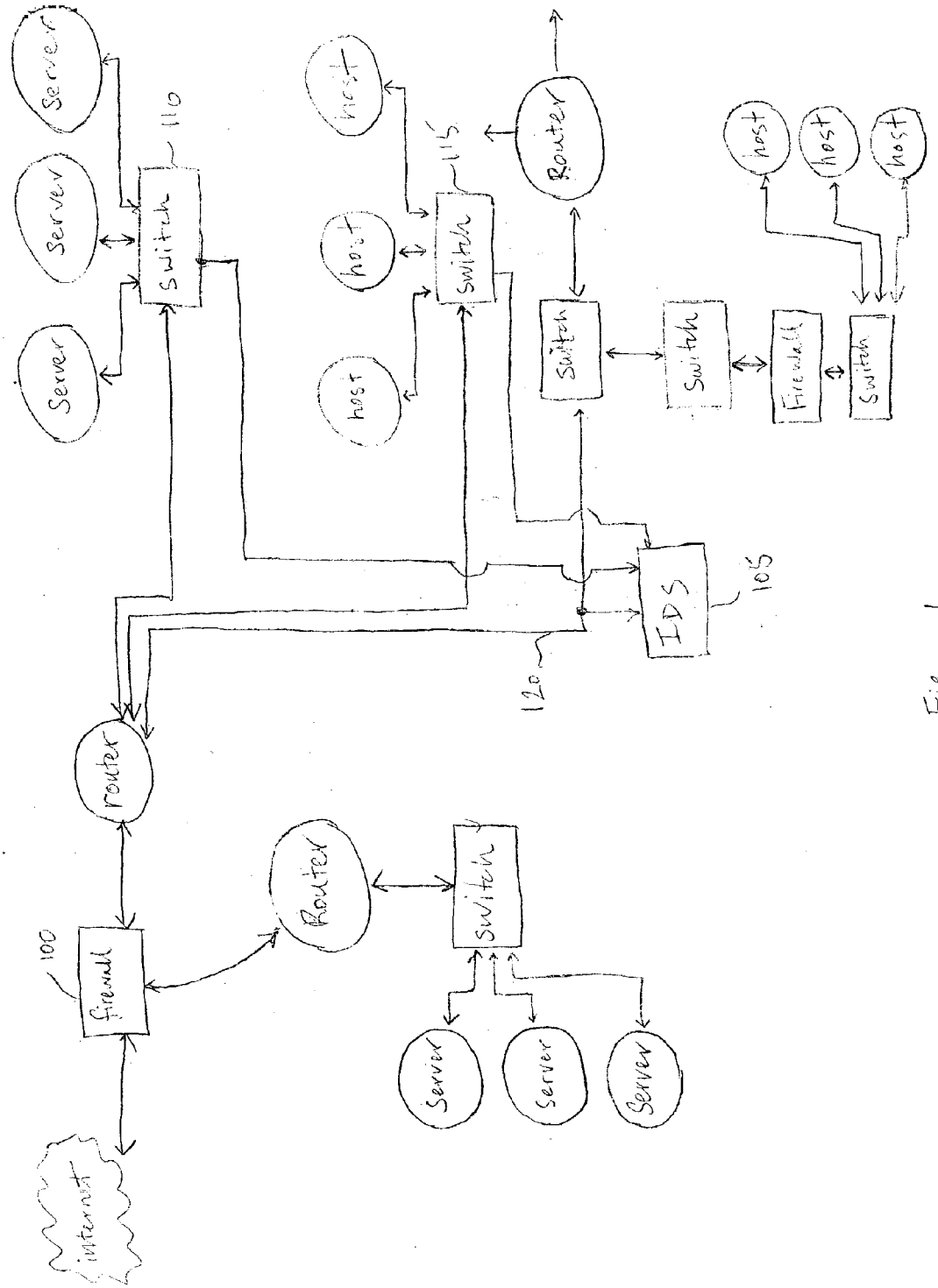

Server
Server
Server
110
Switch
host
host
host
115
Switch
Router
Switch
Switch
Firewall
Switch
host
host
host
IDS
105
120
router
Router
Switch
100
Firewall
Server
Server
Server
internet

Fig. 1

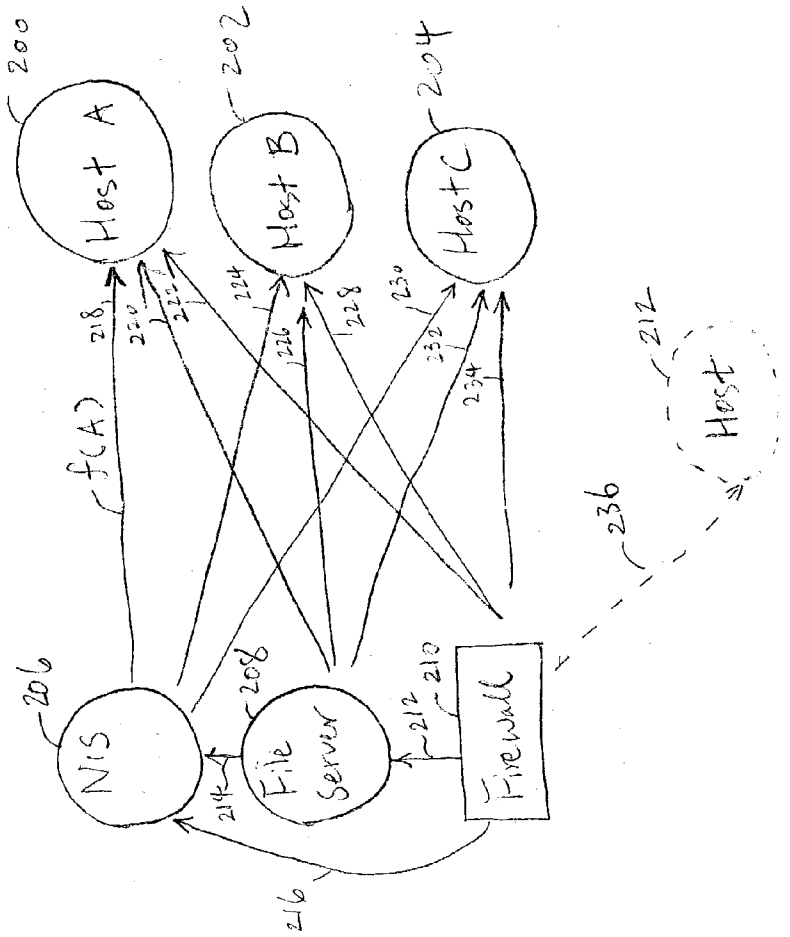

200
Host A
202
Host B
204
Host C
f(A)
218
220
222
224
226
228
230
232
234
206
NIS
208
File Server
210
Firewall
212
Host
214
216
236

Fig. 2

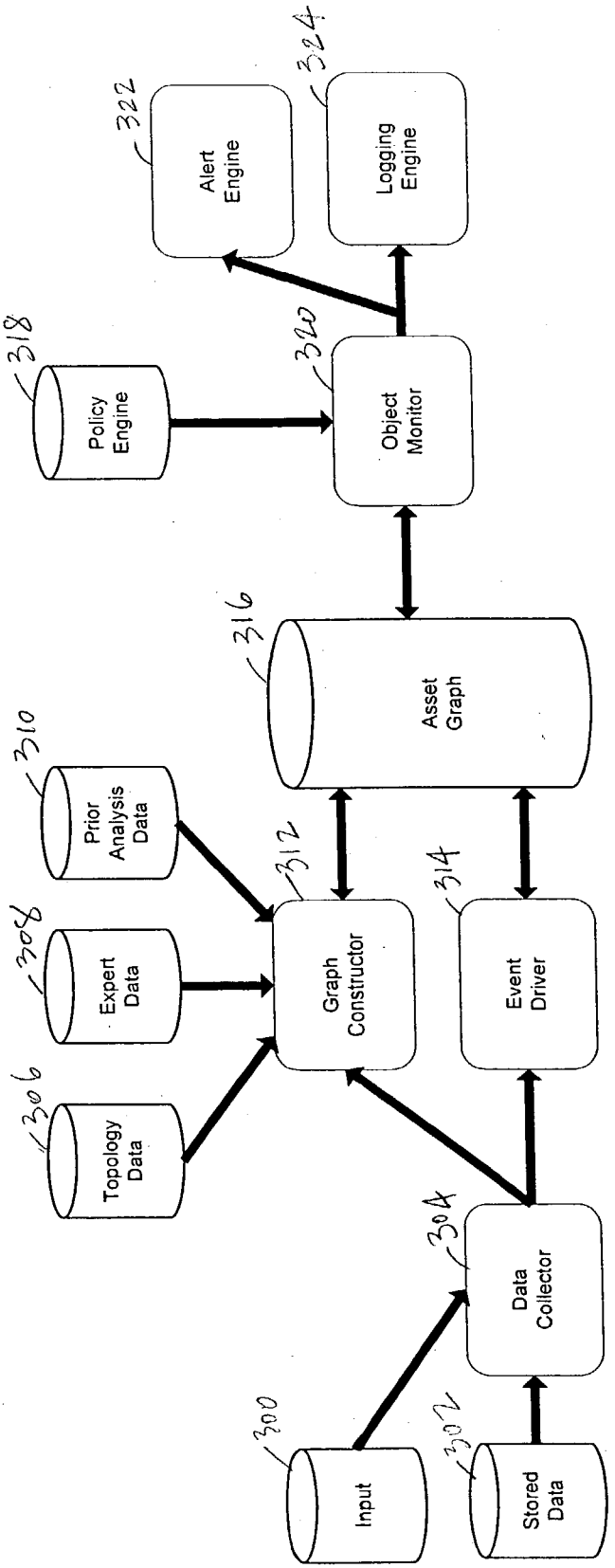

Alert Engine
322
Logging Engine
324
Policy Engine
318
Object Monitor
320
Asset Graph
316
Prior Analysis Data
310
Expert Data
308
Topology Data
306
Graph Constructor
312
Event Driver
314
Data Collector
304
Input
300
Stored Data
302

Fig 3

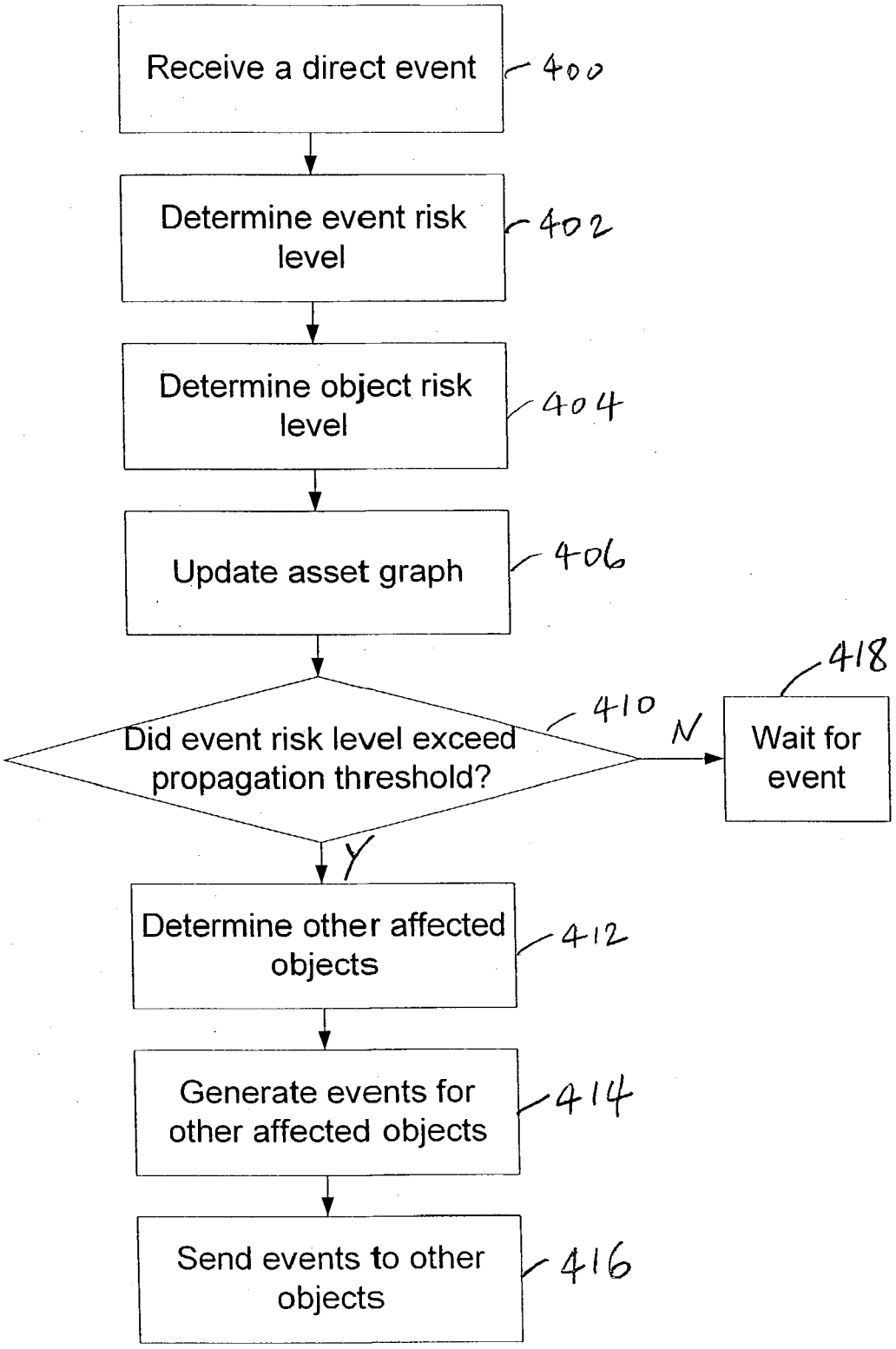

Receive a direct event
400
Determine event risk level
402
Determine object risk level
404
Update asset graph
406
Did event risk level exceed propagation threshold?
410
N
Wait for event
418
Y
Determine other affected objects
412
Generate events for other affected objects
414
Send events to other objects
416

Fig. 4

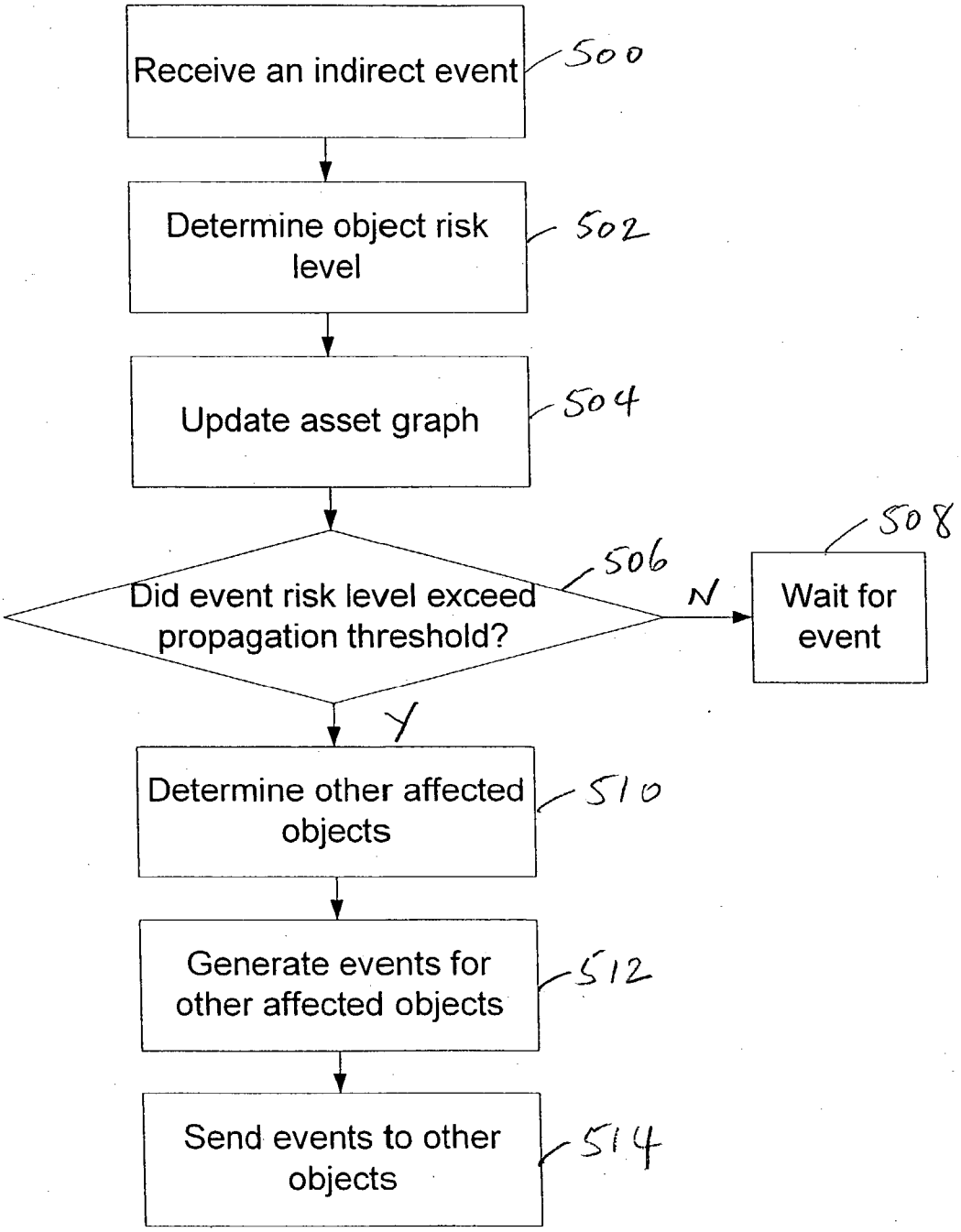

Receive an indirect event
500
Determine object risk level
502
Update asset graph
504
Did event risk level exceed propagation threshold?
506
N
Wait for event
508
Y
Determine other affected objects
510
Generate events for other affected objects
512
Send events to other objects
514

Fig. 5

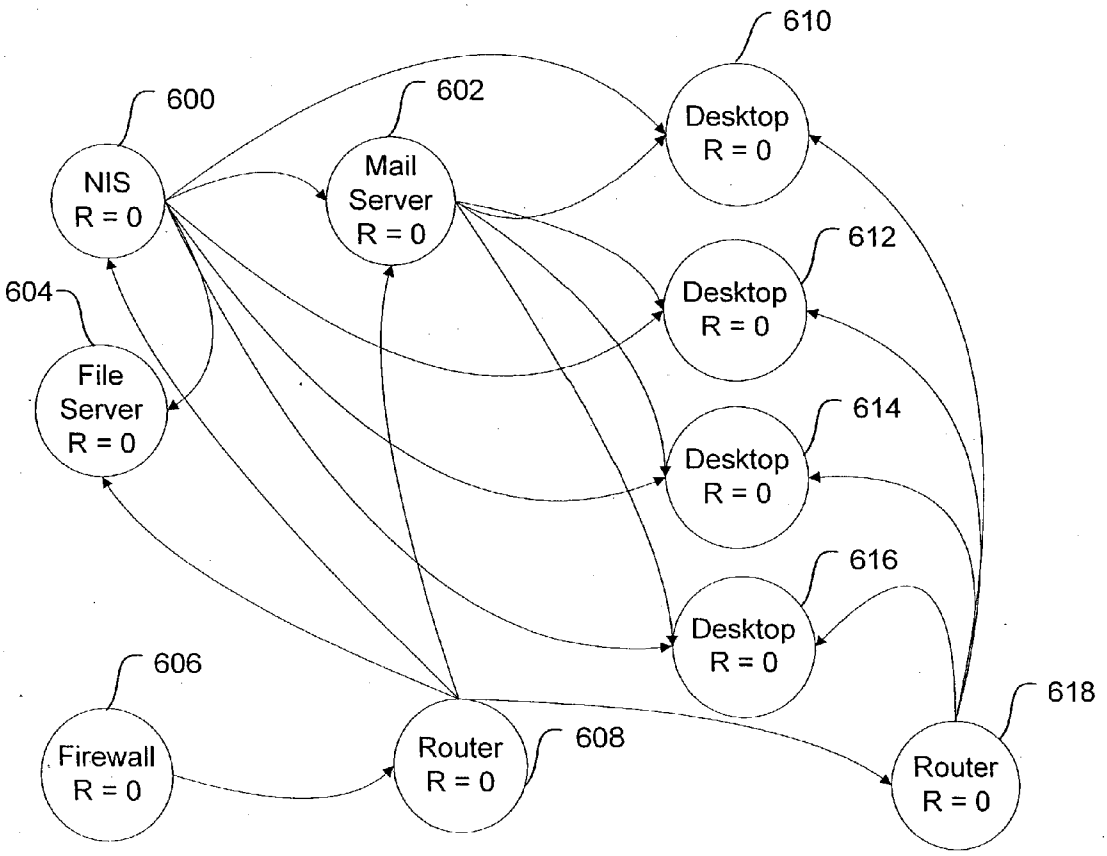

600
NIS
R = 0
602
Mail
Server
R = 0
604
File
Server
R = 0
606
Firewall
R = 0
608
Router
R = 0
610
Desktop
R = 0
612
Desktop
R = 0
614
Desktop
R = 0
616
Desktop
R = 0
618
Router
R = 0

Fig. 6A

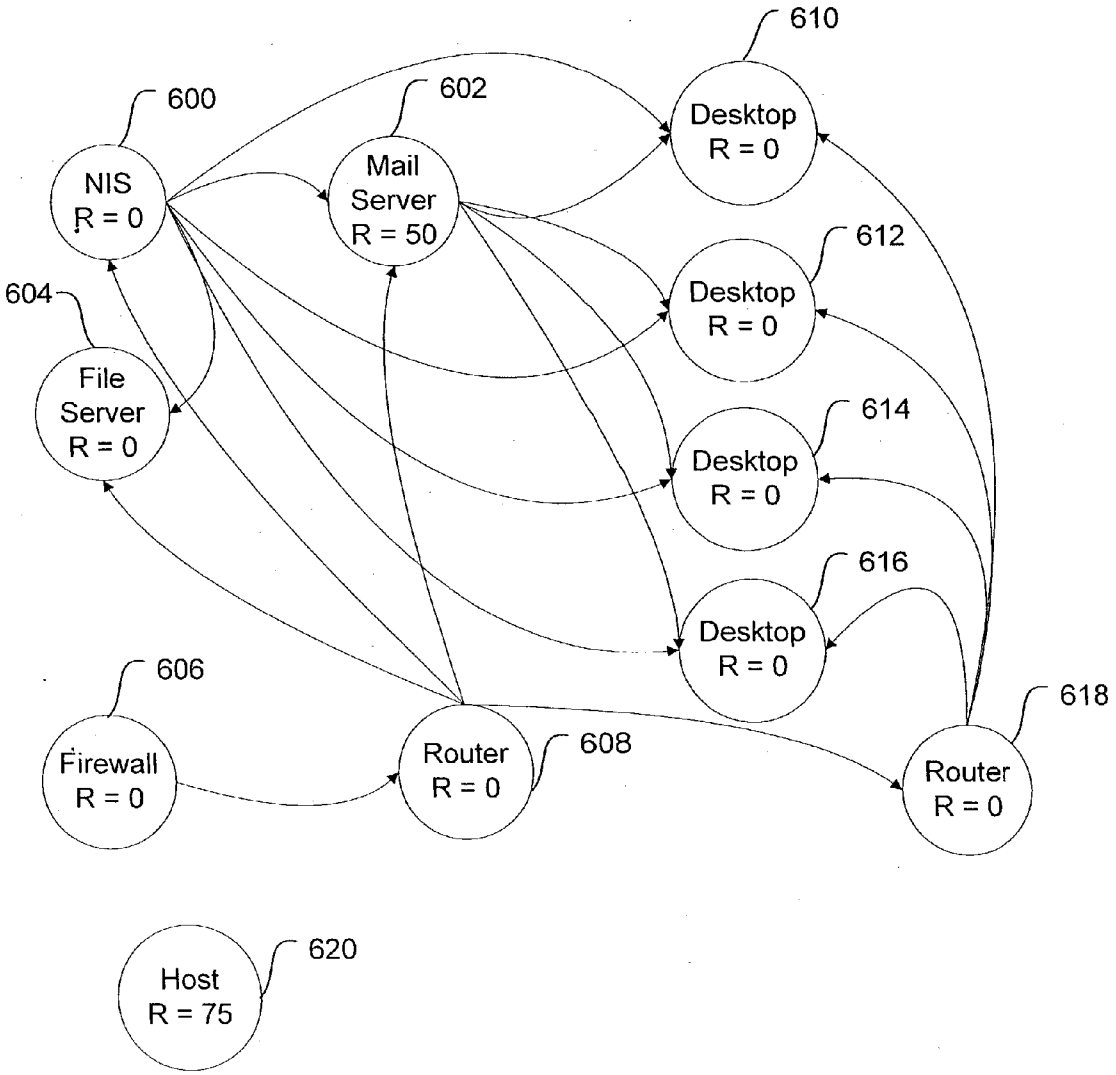

600
NIS
R = 0
602
Mail
Server
R = 50
604
File
Server
R = 0
606
Firewall
R = 0
608
Router
R = 0
610
Desktop
R = 0
612
Desktop
R = 0
614
Desktop
R = 0
616
Desktop
R = 0
618
Router
R = 0
620
Host
R = 75

Fig. 6B

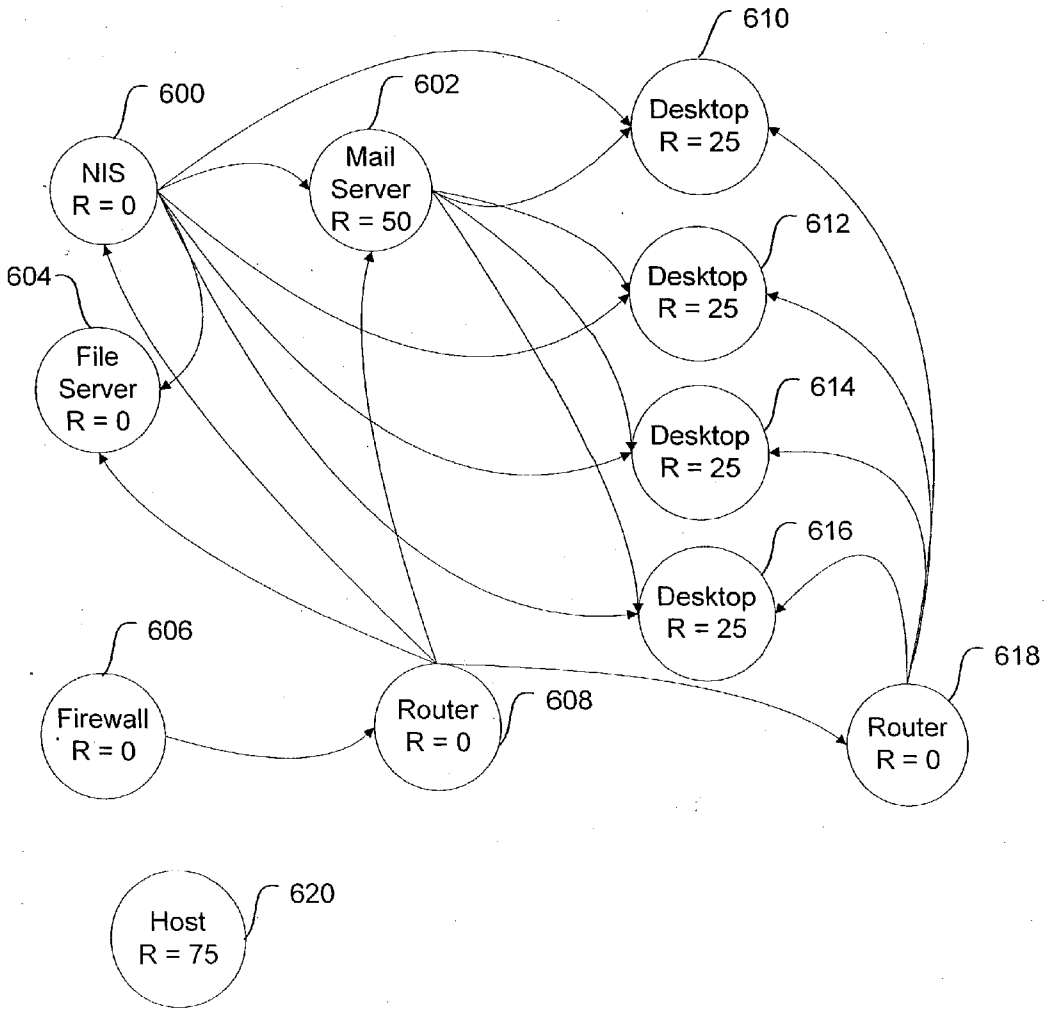

600
NIS
R = 0
602
Mail
Server
R = 50
604
File
Server
R = 0
606
Firewall
R = 0
608
Router
R = 0
610
Desktop
R = 25
612
Desktop
R = 25
614
Desktop
R = 25
616
Desktop
R = 25
618
Router
R = 0
620
Host
R = 75

Fig. 6C

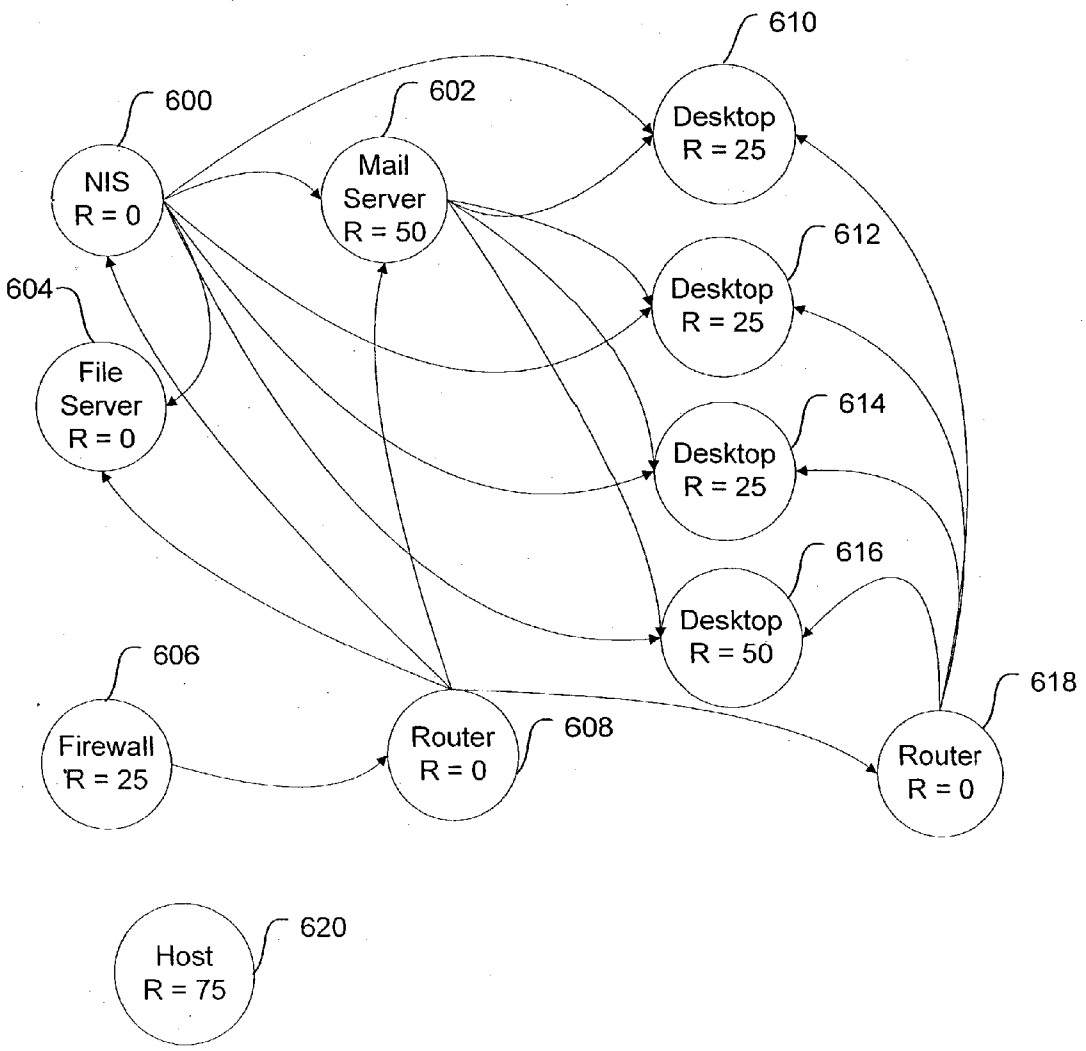

600
NIS
R = 0
602
Mail
Server
R = 50
604
File
Server
R = 0
606
Firewall
R = 25
608
Router
R = 0
610
Desktop
R = 25
612
Desktop
R = 25
614
Desktop
R = 25
616
Desktop
R = 50
618
Router
R = 0
620
Host
R = 75

Fig. 6D

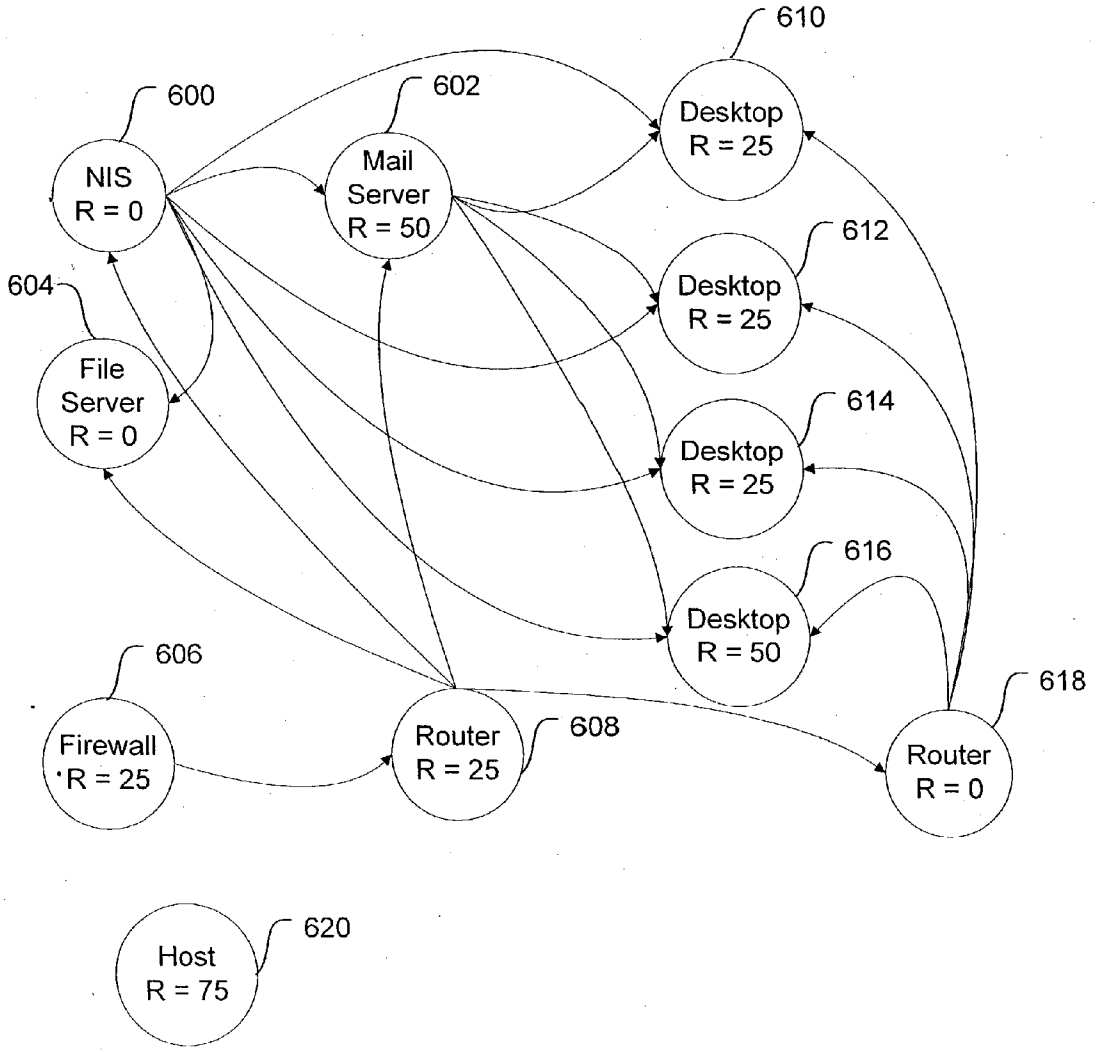

600
NIS
R = 0
602
Mail
Server
R = 50
604
File
Server
R = 0
606
Firewall
R = 25
608
Router
R = 25
610
Desktop
R = 25
612
Desktop
R = 25
614
Desktop
R = 25
616
Desktop
R = 50
618
Router
R = 0
620
Host
R = 75

Fig. 6E

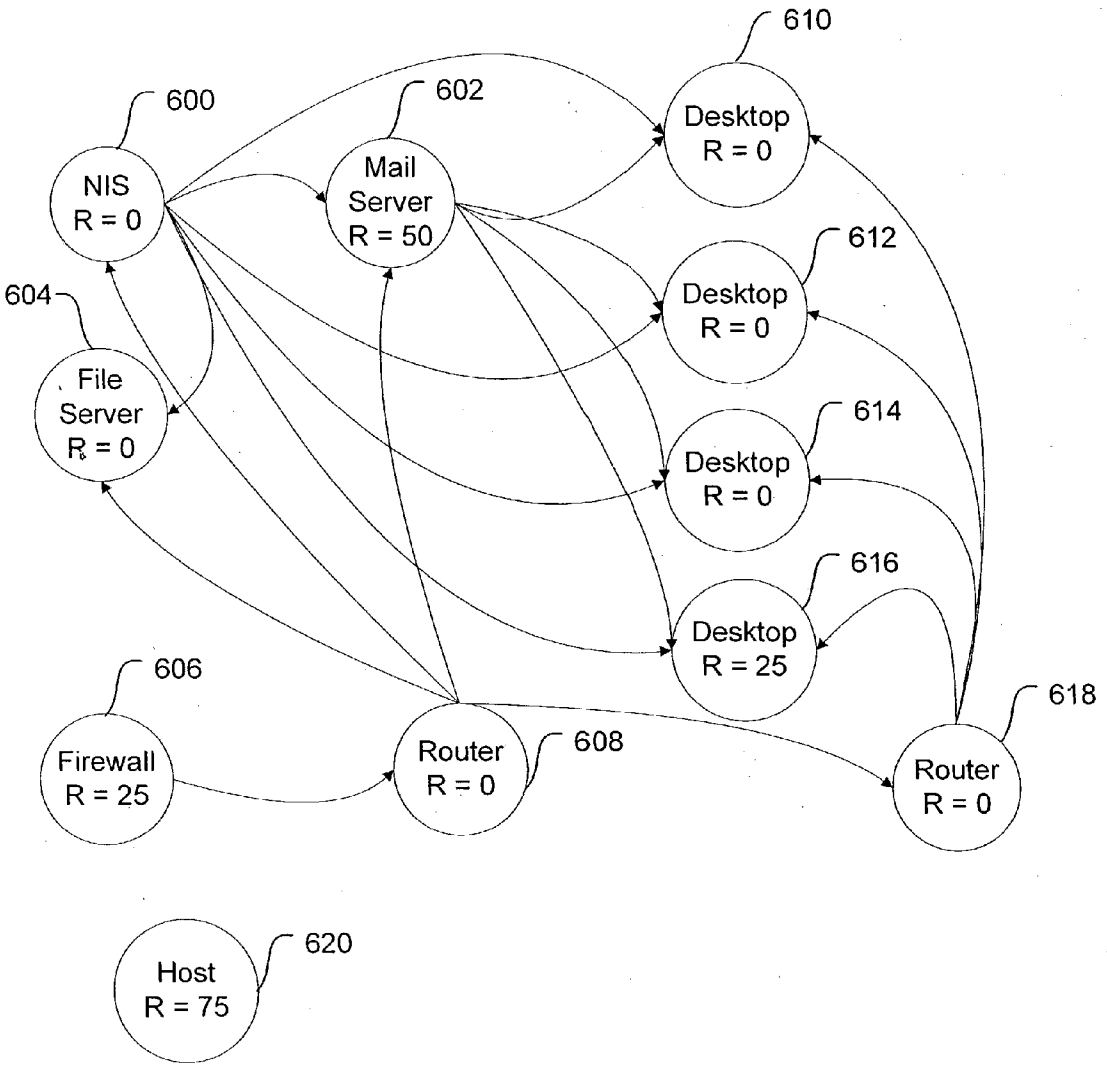

600
NIS
R = 0
602
Mail
Server
R = 50
604
File
Server
R = 0
606
Firewall
R = 25
608
Router
R = 0
610
Desktop
R = 0
612
Desktop
R = 0
614
Desktop
R = 0
616
Desktop
R = 25
618
Router
R = 0
620
Host
R = 75

Fig. 6F

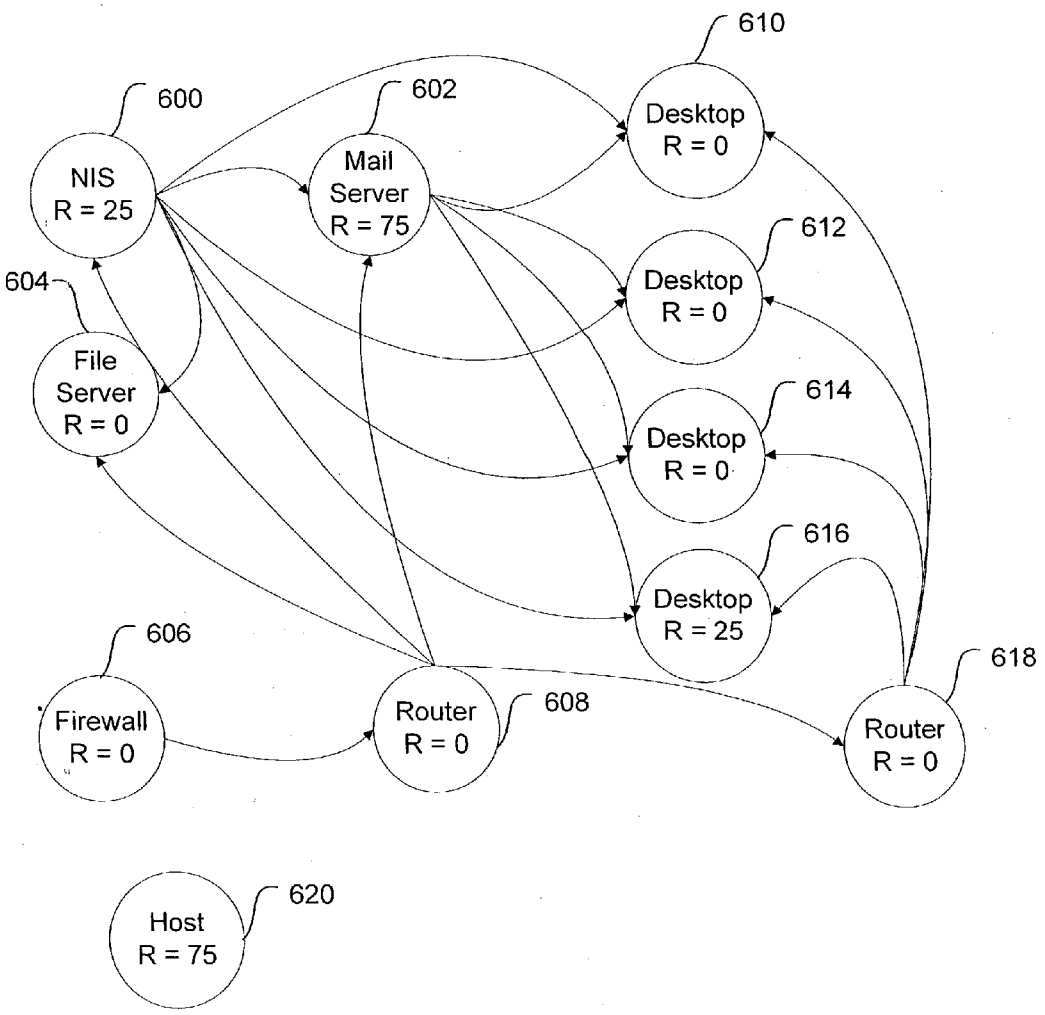

600
NIS
R = 25
602
Mail
Server
R = 75
604
File
Server
R = 0
606
Firewall
R = 0
608
Router
R = 0
610
Desktop
R = 0
612
Desktop
R = 0
614
Desktop
R = 0
616
Desktop
R = 25
618
Router
R = 0
620
Host
R = 75

Fig. 6G

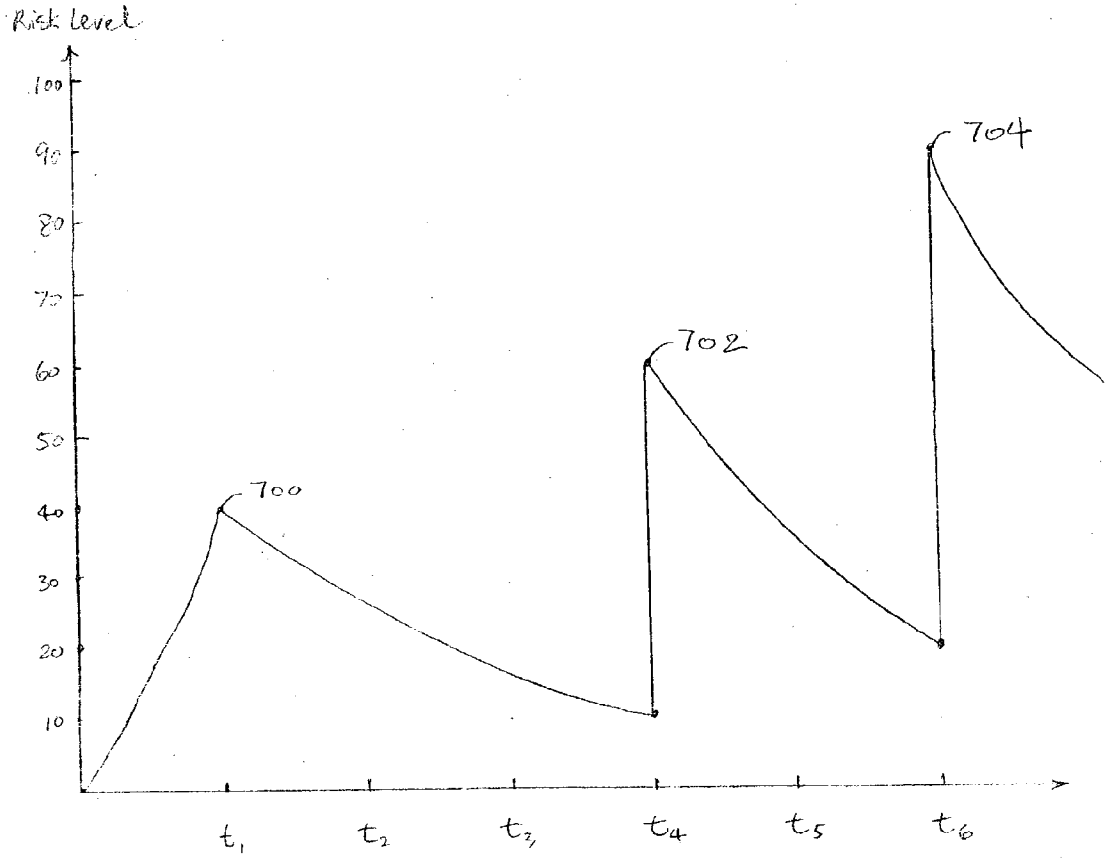


Fig. 7A

Time	risk level	source	destination	type
t_1	40	object A	object B	140
t_4	50	object C	object A	3365
t_6	70	object D	object A	1180

Fig. 7B

NETWORK RISK ANALYSIS

FIELD OF THE INVENTION

[0001] The present invention relates generally to analysis of security networks. More specifically, a method and system for security network analysis based on asset analysis is disclosed.

BACKGROUND OF THE INVENTION

[0002] Computer security is an important aspect of many computer networks. As businesses and institutions become more connected and dependent on the Internet, the risk of attacks has also grown. Most of the intrusion detection systems (IDS) on the market today are capable of collecting different events on the network and displaying them.

[0003] **FIG. 1** is a topology diagram of an example network. A firewall **100** protects the network from the Internet. Different sections of the network are connected to each other and to the Internet via various networking devices on the network, including routers, switches, firewalls, etc. A number of servers and hosts are connected directly or indirectly to some of these networking devices. An IDS **105** collects events associated with a part of the network by connecting with certain networking devices such as switches **110** and **115**, or by tapping into connection links such as link **120**. The IDS may also aggregate the data collected into some central location for later retrieval and analysis, as well as provide interface for accessing the data.

[0004] A security administrator or an authorized user can use the information collected from the network to determine the nature of any suspicious activities. For example, the security administrator may identify certain critical events, examine the data collected to trace the chain of events that are associated with the critical events in order to determine where the threats come from, and then decide what responsive actions to take. Making a reasonable assessment is a time and resource intensive process, and the amount of effort spent is very much dependent on the security administrator's knowledge and experience. Also, as the network becomes more complex, the amount of data collected can be too overwhelming to be processed manually. It would be useful to have a system that can facilitate the assessment process with greater intelligence, thereby reducing the amount of time and resource needed by the process.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

[0006] **FIG. 1** is a topology diagram of an example network.

[0007] **FIG. 2** is an asset graph of an example network according to one embodiment of the present invention.

[0008] **FIG. 3** is a block diagram illustrating one embodiment of a system according to the present invention.

[0009] **FIG. 4** is a flowchart illustrating how a new direct event is processed according to one embodiment of the present invention.

[0010] **FIG. 5** is a flowchart illustrating how a new indirect event is processed according to one embodiment of the present invention.

[0011] FIGS. **6A-6G** are a set of diagrams illustrating the evolution of risk levels of an asset graph over a sample time period, according to one embodiment of the present invention.

[0012] **FIG. 7A** is a graph illustrating the risk level of an example object A over a time period.

[0013] **FIG. 7B** is a table listing some of the properties of the events at t1, t3 and t4, including risk levels, source and destination objects, and types of event.

DETAILED DESCRIPTION

[0014] It should be appreciated that the present invention can be implemented in numerous ways, including as a process, an apparatus, a system, or a computer readable medium such as a computer readable storage medium or a computer network wherein program instructions are sent over optical or electronic communication links. It should be noted that the order of the steps of disclosed processes may be altered within the scope of the invention.

[0015] A detailed description of one or more preferred embodiments of the invention are provided below along with accompanying FIGURES that illustrate by way of example the principles of the invention. While the invention is described in connection with such embodiments, it should be understood that the invention is not limited to any embodiment. On the contrary, the scope of the invention is limited only by the appended claims and the invention encompasses numerous alternatives, modifications and equivalents. For the purpose of example, numerous specific details are set forth in the following description in order to provide a thorough understanding of the present invention. The present invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the present invention is not unnecessarily obscured.

[0016] A method and system of analyzing security risk in a computer network is disclosed. Risk levels of objects (also referred to as assets) associated with events are evaluated. An event is sometimes propagated to other related objects. The risk levels of objects may change over time. Asset relationships are constructed based on the interactions between objects and their dependencies. In some embodiments, an asset graph is used to represent the asset relationships. Techniques for tracing events are also disclosed.

[0017] The topology diagram shown in **FIG. 1** shows physical links between devices. While it is useful to illustrate the topology of the network, it is often not the most effective way of representing the relationships among network objects in the context of security analysis. For instance, an authentication server and a host that interacts with the server may not have a direct network connection and may be physically remote, but an attack on the server can have a direct impact on the host so that the host and the server, from a security standpoint, are logically adjacent. The present invention constructs asset relationships among the objects on the network, and uses the asset relationships

to help analyze the risk levels (also referred to as suspicion level) associated with the objects and events.

[0018] An object (also referred to as assets, network objects, nodes, or elements) comprises an entity on the network that has relevance in terms of security analysis. An object can be, and is not limited to, a physical device (such as a server, a desktop, a laptop, a router, a switch, a firewall, etc.), a software component (such as a web server process, a file, etc.), a user, as well as combinations thereof. An asset relationship between two objects exists when the events that occur on the first object affect the second object; in other words, there is an asset relationship when a dependency exists between the two objects. A suspicious event that has occurred on one object is often propagated to its primary dependent objects or targets, which may also propagate the event to other related objects. There may also be mutual dependency in some cases, where the events that occur on two objects influence each other.

[0019] The objects and their asset relationships can be constructed using a variety of methods, including dynamic analysis based on an input data stream received by the system from an event detection system or other source of events. For example, when the input is analyzed, the analysis may show that there is a source object and a destination object carrying out certain functions. It can then be derived that these two objects (the source and the destination) are in the system and there is an asset relationship between the two. In some embodiments, the assets and their asset relationships are constructed using topology discovery based on the topology of the network, or administrator or expert input that specifies the objects and relationships. In some embodiments, an asset graph is used to construct or present the logical asset relationships.

[0020] **FIG. 2** is an asset graph of an example network according to one embodiment of the present invention. The objects of the network include hosts A **200**, B **202**, C **204**, a network information server (NIS) **206**, a file server **208**, and a firewall **210**. The asset relationships between the objects are represented as arrows pointing in the direction of event propagation. For instance, NIS **206** and file server **208** are dependent on firewall **210**, and are linked by arrows **216** and **212**, respectively. The NIS is also dependent on the file server; therefore, these two objects are linked by an arrow **214**. Hosts A, B and C are dependent on NIS **206** and are linked to NIS **206** by arrows **218**, **220** and **222**, respectively. Hosts A, B and C are dependent on file server **208** and are connected to file server **208** by arrows **224**, **226** and **228**, respectively. Hosts A, B, and C are dependent on firewall **210** and are connected to firewall **210** by arrows **230**, **232**, and **234**, respectively. Since all hosts on the network are dependent on the firewall, this relationship is represented generally by a dashed arrow **236** pointing to a generic host object **212**. It should be noted that in other embodiments, bi-directional arrows may be used to represent objects that are dependent on each other.

[0021] The object risk level of an object is calculated based on a variety of factors including the attribute information of the object (including object type, location, trust relationships, vulnerability state, etc.), events that have been detected at the object, events from other objects on which this object depends, and time dependent functions, etc.

[0022] The attribute information includes the initial, or steady state, object risk level of the object. An incoming event can change the object risk level. The change in object risk level is dependent on factors including the event type, the severity of the event, the source of the event, etc. Also, the object risk level is often time dependent. In one embodiment, when a suspicious event first occurs, the object risk level can be fairly high for the period immediately following the event. As time goes by, the object risk level associated with that event drops. For example, when it is detected that somebody from the outside is probing a server on the network, it is quite likely that there may be attacks or other harmful activities following the probing event, and thus security needs to be tightened. However, as time passes and the anticipated activities do not take place, it may be concluded that the threat has subsided. It would make sense to gradually reduce the object risk level over time, until the increase in risk level due to the probing event is diminished. There are also events that maintain a constant level of threat until further action is taken, as well as events that have an increasing effect on risk level over time if they are not countered or if they persist.

[0023] The object risk level calculation for an object is often generalized as a function or a set of functions. There are many different functions that are used to represent the asset relationships in different embodiments. For instance, in the embodiment shown, the object risk function associated with asset relationship **218** between the NIS and host A, calculated at time t is characterized as the following:

$$R_t = R_{t-1} + g(r, s) + d(t - t_0)$$

[0024] where R_t is the object risk level to be calculated, R_{t-1} is the object risk level of host A at last calculation, g(r, s) is the event risk level of any event that just took place and either directly affects the object being updated or indirectly affects the object being updated as a result of propagation from another object, and $d(t-t_0)$ is a time function that changes the risk level over time. The time function changes the risk of the object regardless of the origin of the risk.

[0025] The term g(r, s) represents the event risk level, where r denotes the intrinsic risk that depends the event's type and intensity, and s denotes the dependency on the source or object from which the event originated. The value of r may be assigned by the system or by the device that captured the event. For example, a login event is determined to pose a smaller level of threat than a port scanning event and thus the login event has a lower event risk level. The value of s is affected by the type of source and the type of connection between the source object and the object receiving the event. For example, a login event that is initiated from an object inside the firewall has less risk than a login event that is initiated from an object outside the firewall. S may be further influenced by the type of connection between the device that was the target of the event and the device that receives the propagated event from the target object. In other embodiments, the event risk level term may also depend on other factors that are deemed to be relevant. In some embodiments, certain types of events may reduce risk instead of increasing risk. When the risk level of the object is being updated, but no event has been sent to the object, g(r, s) equals 0.

[0026] In the term $d(t-t_0)$, t_0 marks the time the object risk level was last updated. In one embodiment, d is an expo-

nential decay function based on time and previous risk level, $R_{t-1} \cdot e^{-y \cdot (t-t_{n-1})}$. If the object risk level is updated regularly, then the exponential decay can be implemented with less computation by using an approximation function such as regularly reducing the risk level by a given percentage or decay factor. In some embodiments, the risk level is a non-exponential decay function. In some embodiments, the risk level is a constant. In some embodiments, the risk level increases over time. In certain embodiments, individual decay functions may be defined for each risk component to allow for different decay rates for risks caused by different types of events.

[0027] Tables 1-5 are some examples of objects used in one embodiment of the asset graph and their attributes. Table 1 is a generic object with a basic set of attributes that are included with all the objects, whereas Tables 2-5 are specific objects used in the asset graph. Some of the attributes are complex attributes comprising multiple values. In other embodiments, the attributes and the objects are different.

TABLE 1

General Object

Attributes	Attribute Details
Name	
Unique ID	
Suspicion Level	
Triggers	
Last updated time stamp	

[0028]

TABLE 2

Host Object

Attributes	Attribute Details
Network address	List of addresses of host
Host name	List of symbolic host names
Class	Type of machine (server, desk top, etc)
Operating system	Type, version, patch level
Services	List of known exposed services and vulnerability state
Trust relationships	List of known hosts that trust and are trusted by this host

[0029]

TABLE 3

Service Object

Attributes	Attribute Details
Ports	
Vulnerabilities	
Owner	
Security impact	

[0030]

TABLE 4

User Object

Attributes	Attribute Details
Name	
Unique ID	User id
Privilege level	
Security impact	

[0031]

TABLE 5

File object

Attributes	Attribute Details
Path	
Type	Data, executable, etc.
Privilege level	
Checksum	
Security impact	

[0032] **FIG. 3** is a block diagram illustrating one embodiment of a system according to the present invention. A data collector **304** collects event data from a variety of sources, including incoming data input **300** and stored data **302**. Thus, the system can both dynamically analyze event data in real time by collecting incoming input data, and analyze event data for post processing by using stored data. A graph constructor **312** constructs an asset graph **316** using different types of input information including topology data **306** that includes the topology of the network, expert data **308** that is input by someone who is knowledgeable of the network, prior analysis data **310**, as well as output from data collector **304**.

[0033] Data collector **304** also sends the collected event to an event driver **314**, which analyzes the event data, and updates the asset graph if the event introduces changes in the objects, their attributes or their asset relationships. In addition, the event driver propagates the event according to the risk propagation functions associated with the objects in the event, and tracks the effects of the event. An object monitor **320** monitors asset graph **316** for objects that have high object risk levels. The object monitor takes inputs from policy engine **318**, and reacts according to the policies to objects whose object risk levels have reached their triggering levels. The object monitor communicates with an alert engine **322** to generate an alert, and a logging engine **324** to log the event. In some embodiments, the object monitor also communicates with mail servers to generate email messages or send pages.

[0034] An event may be either a direct event that directly affects specific objects targeted by the event, or a propagated event that affects objects other than the target objects and have been propagated as a result of asset relationships between objects.

[0035] **FIG. 4** is a flowchart illustrating how a new direct event is processed according to one embodiment of the present invention. A direct event is received by the system

(**400**). An event can come from a variety of sources, including intrusion detection systems, network management systems, etc. In some embodiments, the event has an event risk level that is provided by the device that detects the event. In other embodiments, the event is reported without an event risk level by the detecting device, and the event risk level is calculated upon receiving the event. Even if an event risk level is provided by the detecting device, different detecting devices may use different scales for assigning risk levels to an event—an event risk level of 7 on a scale of 1-10 is significantly different from a risk level of 7 on a scale of 1-50. To achieve consistency, the event risk level is either normalized to produce canonical values that are arranged on the same scale, or calculated directly by the system (**402**).

[**0036**] Generally, an event may have more than one target object, that is, it may directly affect more than one object. The primarily or directly affected objects or targets of the event may include both the source and the destination of an action associated with the event. For example, the primarily affected objects of a telnet session originated from a workstation to a server include both the workstation and the server. The object risk levels of the targets are then updated based on the event risk level, the objects' time functions, the previous object risk levels, etc. (**404**).

[**0037**] The asset graph is then updated according to the object risk levels (**406**). In this embodiment, the object risk levels of the primary targets of the direct event are updated accordingly. If applicable, the event is propagated to related objects that are not directly or primarily affected by the event itself, but whose object risk levels should be adjusted due to the occurrence of the event and their logical association with the object or objects that were the targets of the event. To propagate the event, first it is determined whether the event risk level exceeds a propagation threshold (**410**). In other words, events that are not deemed risky or serious enough to warrant propagation are suppressed.

[**0038**] If the event risk level does not exceed the threshold, the system cleans up and becomes ready to receive events again (**418**). If the event risk level does exceed the threshold, the next level of affected objects is then determined (**412**). Events are then generated for the other affected objects based on the corresponding asset relationships and risk functions (**414**). Eventually, the events are sent to the other affected objects (**416**).

[**0039**] **FIG. 5** is a flowchart illustrating how a new indirect event is processed according to one embodiment of the present invention. Once an indirect event that is propagated from another object is received (**500**), the object risk level of the event recipient is updated directly since the event risk level was previously normalized (**502**). The asset graph is updated according to the object risk levels (**504**), and it is determined whether the event risk level exceeds the propagation threshold (**506**). If the risk level is below the threshold, the system cleans up and becomes ready to receive events again (**508**). If, however, the risk level exceeds the threshold, then other affected objects are determined (**510**). Events are then generated for the other affected objects based on the corresponding asset relationships and risk functions (**512**), and the events are sent to the other affected objects (**514**).

[**0040**] FIGS. **6A-6G** are a set of diagrams illustrating the evolution of risk levels of an asset graph over a sample time period, according to one embodiment of the present invention. **FIG. 6A** illustrates the initial states of the asset graph. The asset graph is constructed based on inputs including topology information provided by a network management tool and information input by an expert. The objects in this asset graph include NIS **600**, mail server **602**, file server **604**, firewall **606**, routers **608** and **618**, plus desktops **610-616**. R represents the canonical risk level of each object on a scale of 0-100. The higher the value, the more suspicious and risky an object is. Initially, the risk levels for all the objects are 0.

[**0041**] **FIG. 6B** shows the primary target evaluation upon receiving a first event. In this embodiment, the event is received from a network intrusion detection system sensor and is presented as the following:

[**0042**] 172.196.10.12:5672→212.44.141.23:25 risk level=75 type=2345 time=9/1/02--19:22:11,

[**0043**] where 172.196.10.12 is the source, and 212.44.141.23 is the destination. The source and destination of this event are the targets of the event. The event has a canonical risk level of 75 on a scale of 0-100. Type is a reference to an internal list of event types. In this embodiment, type **2345** refers to an SMTP buffer overflow attempt. The system does not have information about the source host initially, as the source host is not part of the current asset graph. A new host object **620** is created to represent the source host and added to the asset graph. Since little else is known about this new object, it is marked as highly suspicious with a risk level of 75. The destination object, on the other hand, is known. The destination maps to mail server **602**, which runs an SMTP service on port **25**. Thus, the mail server appears vulnerable to the exploit of the incoming event. Based on this information, the system marks the mail server as suspicious, with a risk level of 50.

[**0044**] **FIG. 6C** illustrates the propagation effect as a result of the first event. The first primary target (source object, host **620**) is examined and is determined to have no relationship with the rest of the graph. Thus, no more processing is performed on the source host object. The second primary target (destination object, mail server **602**) is then examined. Desktops **610-616** interact with the mail server **602** and have asset relationships with the mail server. If the mail server is compromised, the desktops are also at risk. The system then propagates a portion of the risk to these desktops, and mark them as somewhat suspicious at a risk level of 25. Each of the desktops is also analyzed to determine whether the risk should be further propagated. For the sake of simplicity, it is assumed that the risk level at this point has fallen below the threshold and ceases to propagate.

[**0045**] **FIG. 6D** illustrates how a second event is processed in the embodiment shown. Shortly after the first event was received, a second new event is received, indicating that desktop **616** (source object) is port scanning firewall **606** (destination object). The risk levels of the targets are each increased by 25. Firewall **606** now has a risk level of 25, and desktop **616** now has a risk level of 50.

[**0046**] **FIG. 6E** illustrates the propagation of the event received in **FIG. 6D**. Router **608**, which has an asset relationship with firewall **606**, has an increase in its risk level as a result of the port scanning event being propagated.

[0047] **FIG. 6F** illustrates the effect of the passage of time has on the embodiment shown. As time passes, the events become less relevant. The objects are reevaluated and their risk levels are reduced. All desktops except for **616** have fallen back to their original default value. However, **616** is still marked somewhat suspicious as a result of the port scanning event. Firewall **606** is still somewhat suspicious at a risk level of 25. Host **620** and mail server **602** still have fairly high risk levels at 75 and 50, respectively.

[0048] **FIG. 6G** illustrates the effects of a third new event. In this case the new event indicates possible port scanning activity of NIS **600** originating from the mail server. While such activity normally may not warrant an alert to be generated, when taken into account with previous events and the proximity in time, has become a bigger concern. The risk level of mail server **602** is now at 75, which triggers an alert.

[0049] In the example shown above, once mail server **602** triggers the alert, an analysis trail is produced detailing the events leading up to the alert and their effects. The system presents the data to the security administrators using text, graphics, sounds, or a combination thereof. The analysis trail can be produced independent of the risk levels. The analysis trail includes information used for tracing of event effects such as propagation and time. In some embodiments, the analysis trail is used for reconstructing a chain of events that has previously occurred, or a chain of events that leads to an intrusion or causes an object to become highly vulnerable.

[0050] **FIG. 7A** is a graph illustrating the risk level of an example object A over a time period. The risk level is calculated periodically, from t_1 to t_6. At times t_1, t_4 and t_6, three events are received for this object. These events create peaks in risk level at **700**, **702** and **704**, respectively. Between these events, the risk level decreases over time. **FIG. 7B** is a table listing some of the properties of the events at t1, t3 and t4, including risk levels, source and destination objects, and types of event.

[0051] A method and system of analyzing security risk in a computer network has been disclosed. Risk level based on event type and time is used to assess the risk of objects in the network. By using techniques that employ asset relationships and asset graphs, events are better traced and presented. As a result, the amount of time and resource needed in risk management is reduced.

[0052] Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. It should be noted that there are many alternative ways of implementing both the process and apparatus of the present invention. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

What is claimed is:

1. A method of analyzing security risk in a computer network comprising:

constructing asset relationships among a plurality of objects in the computer network;

receiving an event associated with a selected object; wherein the event has an event risk level;

propagating the event to objects related to the selected object if the event risk level exceeds a propagation threshold.

2. A method of analyzing security risk in a computer network as recited in claim 1 wherein constructing asset relationships includes creating an asset graph.

3. A method of analyzing security risk in a computer network as recited in claim 1 wherein the selected object has a plurality of associated attributes.

4. A method of analyzing security risk in a computer network as recited in claim 1 further comprising determining an object risk level for the selected object.

5. A method of analyzing security risk in a computer network as recited in claim 1 further comprising determining an object risk level for the selected object based on the event received and a time function that changes the object risk level over time.

6. A method of analyzing security risk in a computer network as recited in claim 1 further comprising determining an object risk level for the selected object based on the event received and a time function; wherein the time function is a decay function that decreases the object risk level over time.

7. A method of analyzing security risk in a computer network as recited in claim 1 further comprising determining an object risk level for the selected object based on the event received and a time function; wherein the time function is an exponential decay function that decreases the event risk level over time.

8. A method of analyzing security risk in a computer network as recited in claim 1 further comprising determining an object risk level for the selected object and triggering a reaction once the object risk level reaches a triggering level.

9. A method of analyzing security risk in a computer network as recited in claim 1 further comprising:

determining an object risk level for the selected object; and

triggering a reaction once the object risk level reaches a triggering level;

wherein the reaction includes notifying a user.

10. A method of analyzing security risk in a computer network as recited in claim 1 further comprising:

determining an object risk level for the selected object; and

triggering a reaction once the object risk level reaches a triggering level;

wherein the reaction includes logging the event.

11. A method of analyzing security risk in a computer network as recited in claim 1 further comprising:

determining an object risk level for the selected object; and

triggering a reaction once the object risk level reaches a triggering level;

wherein the reaction includes producing an analysis trail.

12. A method of analyzing security risk in a computer network as recited in claim 1 further comprising:

determining an object risk level for the selected object; and

triggering a reaction once the object risk level reaches a triggering level;

wherein the reaction includes producing an analysis trail comprising information for tracing event effects.

13. A method of analyzing security risk in a computer network as recited in claim 1 further comprising:

determining an object risk level for the selected object; and

triggering a reaction once the object risk level reaches a triggering level;

wherein the reaction includes producing an analysis trail comprising information for tracing event effects and reconstructing a chain of events that occurred previously using the analysis trail.

14. A method of analyzing security risk in a computer network as recited in claim 1 further comprising propagating the event to dependent objects that have a dependency on the selected object; wherein propagating the event to the dependent objects includes determining whether the event has an event risk level that exceeds a propagation threshold.

15. A method of analyzing security risk in a computer network as recited in claim 1 further comprising normalizing the event risk level based on a canonical scale.

16. A method of analyzing security risk in a computer network as recited in claim 1 wherein the selected object is a target object of the event.

17. A method of analyzing security risk in a computer network as recited in claim 1 wherein the event received is real time input or stored data.

18. A method of analyzing security risk in a computer network comprising:

receiving an event associated with a selected object in the computer network;

determining an object risk level for the selected object based on the event received and a time function that changes the object risk level over time.

19. A method of analyzing security risk in a computer network as recited in claim 18 wherein the time function is a decay function that decreases the object risk level over time.

20. A method of analyzing security risk in a computer network as recited in claim 18 wherein the time function is an exponential decay function.

21. A computer program product for analyzing security risk in a computer network, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

constructing asset relationships among a plurality of objects in the computer network;

receiving an event associated with a selected object; wherein the event has an event risk level; and

propagating the event to objects related to the selected object if the event risk level exceeds a propagation threshold.

22. A system for analyzing security risk in a computer network comprising:

an input interface configured to receive an event associated with a selected object;

a processor configured to construct asset relationships among a plurality of objects in the computer network and propagating the event to objects related to the selected object if the event risk level exceeds a propagation threshold.

* * * * *