US008117659B2

(12) **United States Patent**
Hartrell et al.

(10) **Patent No.: US 8,117,659 B2**
(45) **Date of Patent: Feb. 14, 2012**

(54) **MALICIOUS CODE INFECTION CAUSE-AND-EFFECT ANALYSIS**

(75) Inventors: **Gregory D. Hartrell**, Redmond, WA (US); **David J. Steeves**, Seattle, WA (US); **Efim Hudis**, Bellevue, WA (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1023 days.

(21) Appl. No.: **11/321,754**

(22) Filed: **Dec. 28, 2005**

(65) **Prior Publication Data**

US 2007/0150957 A1 Jun. 28, 2007

(51) **Int. Cl.**
G06F 21/00 (2006.01)

(52) **U.S. Cl.** **726/24**; 726/22; 726/23; 726/25; 709/223; 709/224; 713/187; 713/188

(58) **Field of Classification Search** 726/22–30; 713/193, 194, 200, 187–188; 709/227–229, 709/217–222; 715/734–736; 380/2
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,408,642 A * 4/1995 Mann 714/38
7,093,239 B1 * 8/2006 van der Made 717/135
7,832,012 B2 * 11/2010 Huddleston 726/24
2003/0084323 A1 * 5/2003 Gales 713/200
2005/0076262 A1 * 4/2005 Rowan et al. 714/6
2006/0021054 A1 1/2006 Costa et al.
2006/0031933 A1 2/2006 Costa et al.
2006/0090205 A1 * 4/2006 Ellis 726/25
2006/0242705 A1 * 10/2006 Sadhasivam et al. 726/23

OTHER PUBLICATIONS

Guofei Gu, BotHunter : Detecting Malware Infection Trough IDS-Driven Dialog Correlation , Computer Science Laboratory SRI International.*

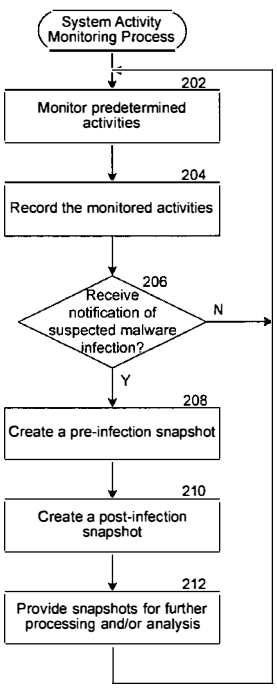
* cited by examiner

Primary Examiner — Taghi Arani
Assistant Examiner — Josnel Jeudy
(74) *Attorney, Agent, or Firm* — Perkins Coie LLP

(57) **ABSTRACT**

A malware analysis system for automating cause and effect analysis of malware infections is provided. The malware analysis system monitors and records computer system activities. Upon being informed of a suspected malware infection, the malware analysis system creates a time-bounded snapshot of the monitored activities that were conducted within a time frame prior to the notification of the suspected malware infection. The malware analysis system may also create a time-bounded snapshot of the monitored activities that are conducted within a time frame subsequent to the notification of the suspected malware infection. The malware analysis system provides the created snapshot or snapshots for further analysis.

20 Claims, 8 Drawing Sheets



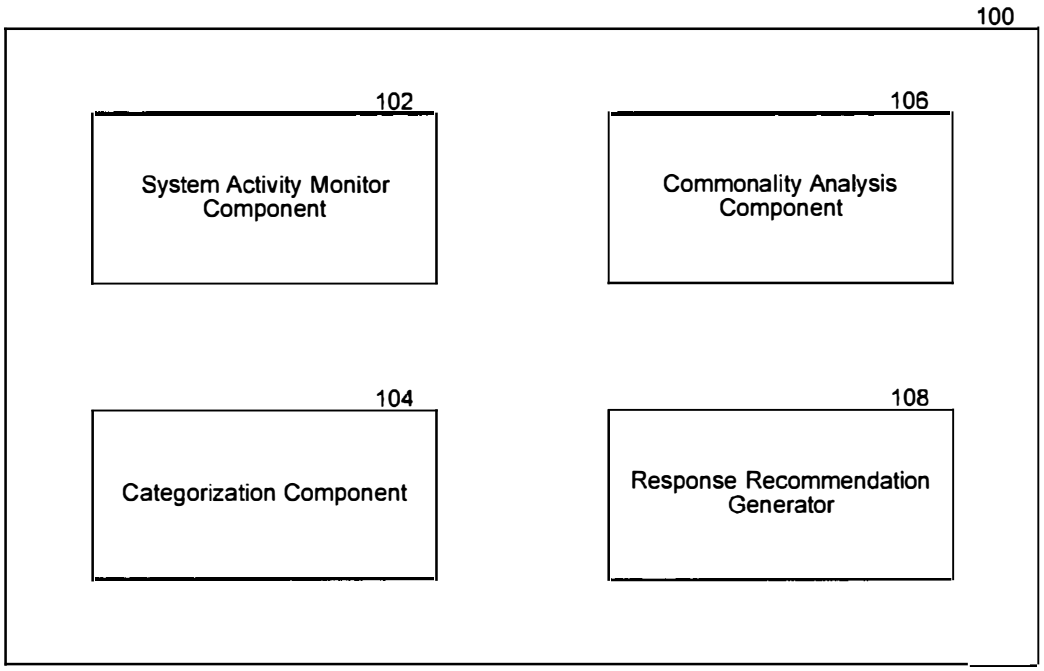

100
102
System Activity Monitor Component
106
Commonality Analysis Component
104
Categorization Component
108
Response Recommendation Generator

FIG. 1

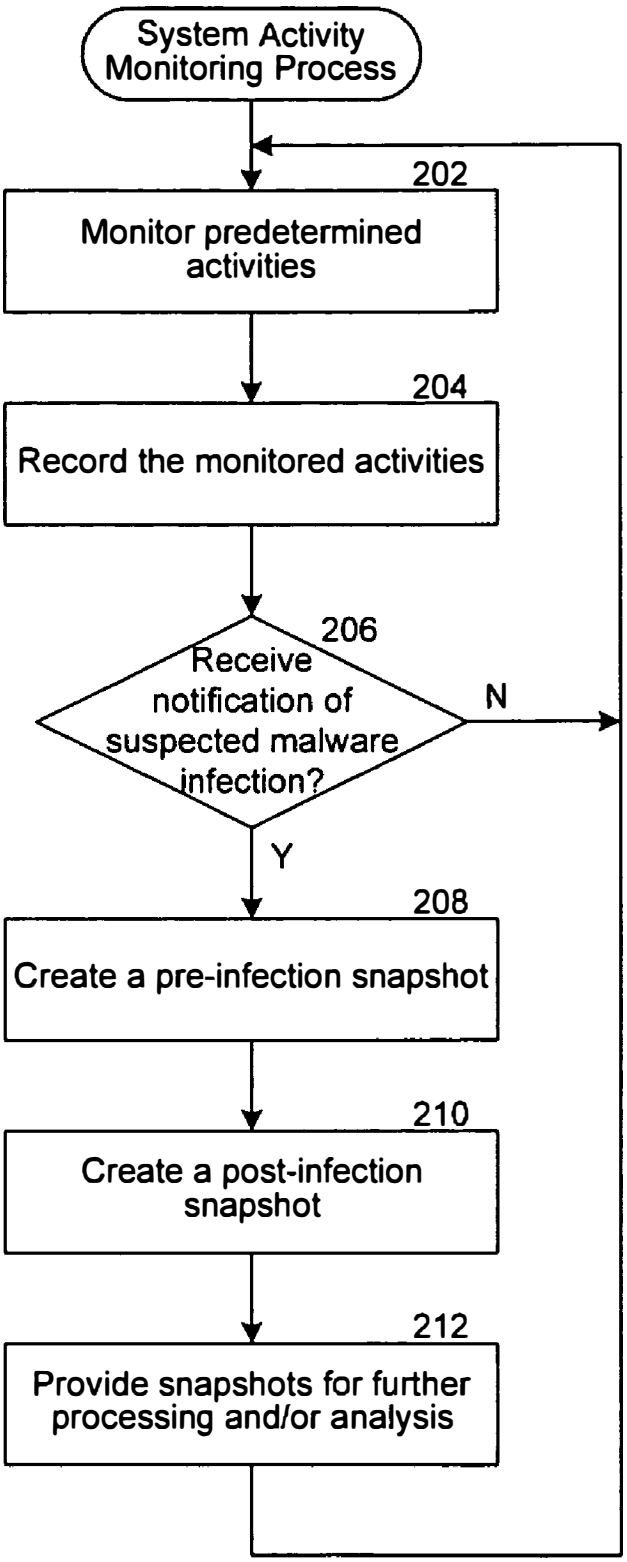

System Activity Monitoring Process
202
Monitor predetermined activities
204
Record the monitored activities
206
Receive notification of suspected malware infection?
N
Y
208
Create a pre-infection snapshot
210
Create a post-infection snapshot
212
Provide snapshots for further processing and/or analysis

FIG. 2

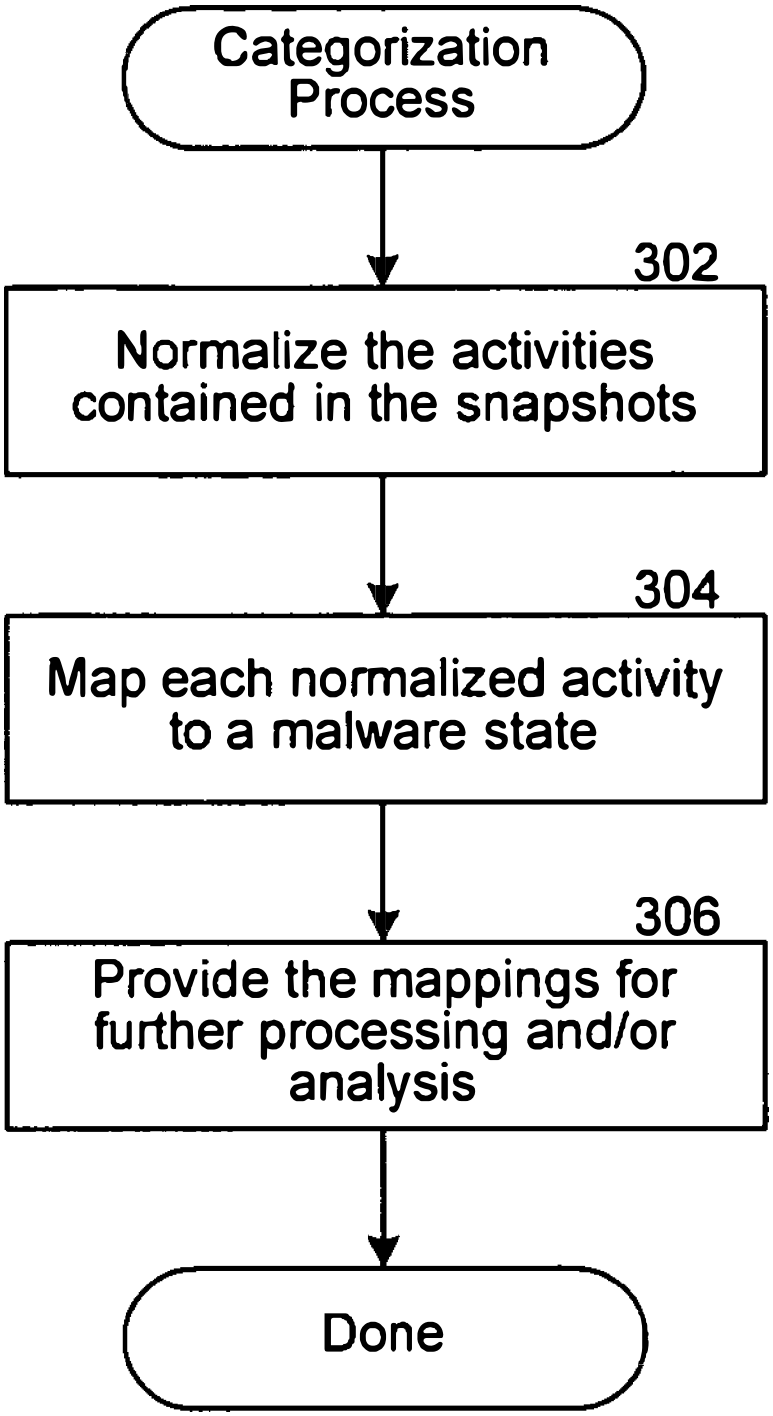

Categorization Process
302
Normalize the activities contained in the snapshots
304
Map each normalized activity to a malware state
306
Provide the mappings for further processing and/or analysis
Done

FIG. 3

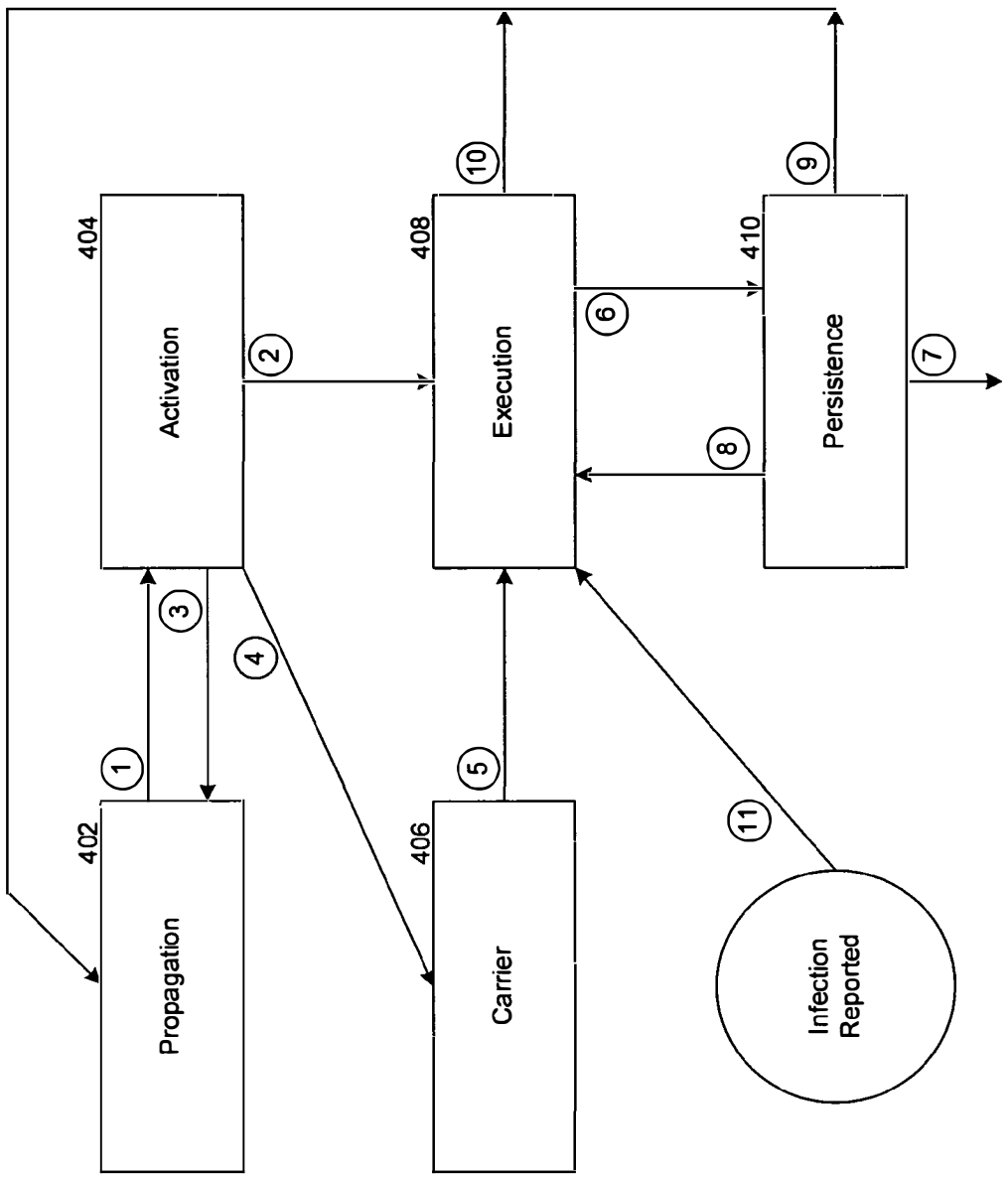

402
Propagation
404
Activation
406
Carrier
408
Execution
410
Persistence
Infection Reported
1
2
3
4
5
6
7
8
9
10
11

FIG. 4

500

502

504 Normalized Action	506 Pre-Notification Malware State	508 Post-Notification Malware State	510 State Transition
File Transfer (e.g., FTP)	Carrier	Persistence	External Communication
Instant Message	Activation	Persistence	External Communication
Remote Authentication	Activation	Persistence	Remote Control
Remote Control Access	Activation	Persistence	Remote Control
Network Facing Service Access	Propagation	Persistence	Remote Control
Outbound Network Connection	Carrier	Propagation	- - - -
Web Site HTTP Get	Activation	Persistence	External Communication
Web File Transfer	Carrier	Execution	Install
Application Launched	Activation	Execution	Command & Control
File Copy from Media	Carrier	Execution	Install
File Created	Carrier	Execution	Install
Configuration Change	Activation	Execution	Command & Control
. . .	. . .	. . .	. . .

FIG. 5

600

602

604 Time	606 Recorded Activity	608 Normalized Action	610 Associated Malware State
t0-2	Inbound TCP connection to port 135 (RPC)	Network Service Access	Activation
t0-1	Outbound UDP connection on port 69 (TFTP)	File Transfer	Carrier
t0	File created (msblast.exe)	File Created	Execution
t0+1	Register key modified (HKLM/. . .)	Configuration Change	Execution
t0+2	Socket opened for listening on port 4444/TCP	New Socket Listing	Persistence
t0+3	Outbound TCP connection to port 135 (RPC)	Outbound Network Connection	Propagation
t0+4	Outbound TCP connection to port 135 (PPC)	Outbound Network Connection	Propagation
...	. . .	. . .	. . .

FIG. 6

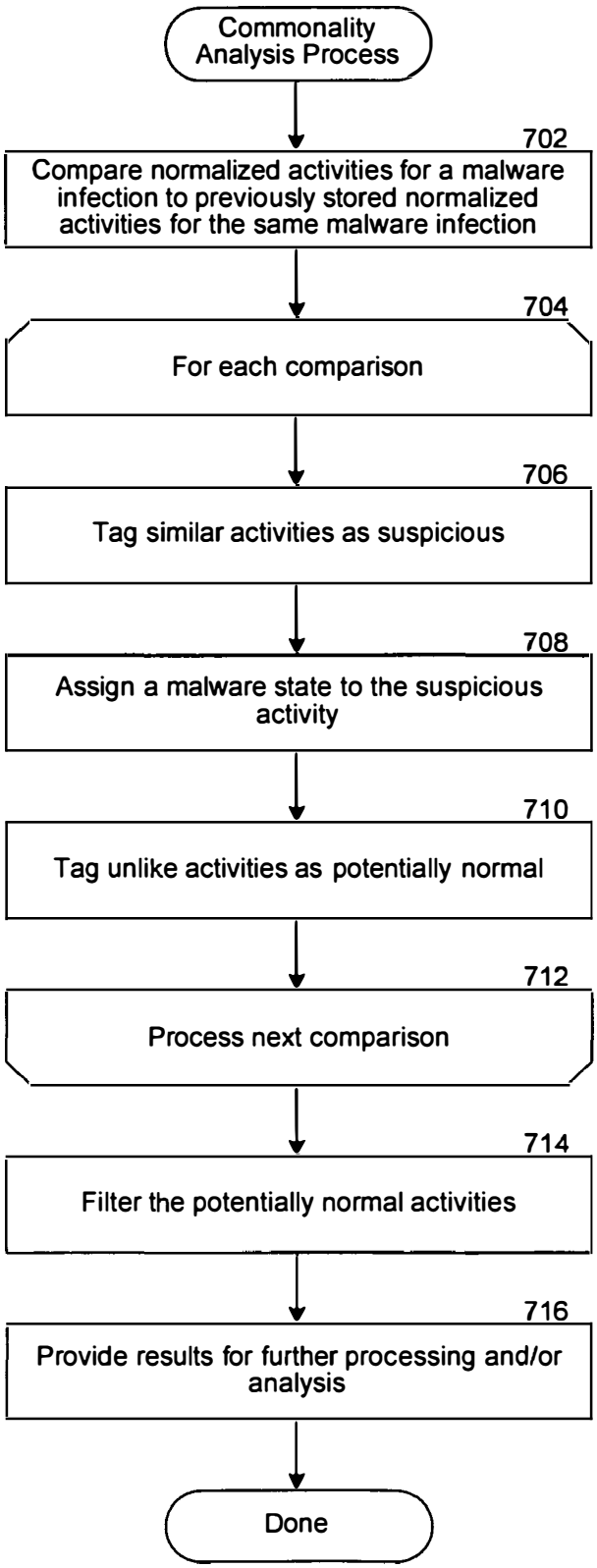

Commonality Analysis Process
702
Compare normalized activities for a malware infection to previously stored normalized activities for the same malware infection
704
For each comparison
706
Tag similar activities as suspicious
708
Assign a malware state to the suspicious activity
710
Tag unlike activities as potentially normal
712
Process next comparison
714
Filter the potentially normal activities
716
Provide results for further processing and/or analysis
Done

FIG. 7

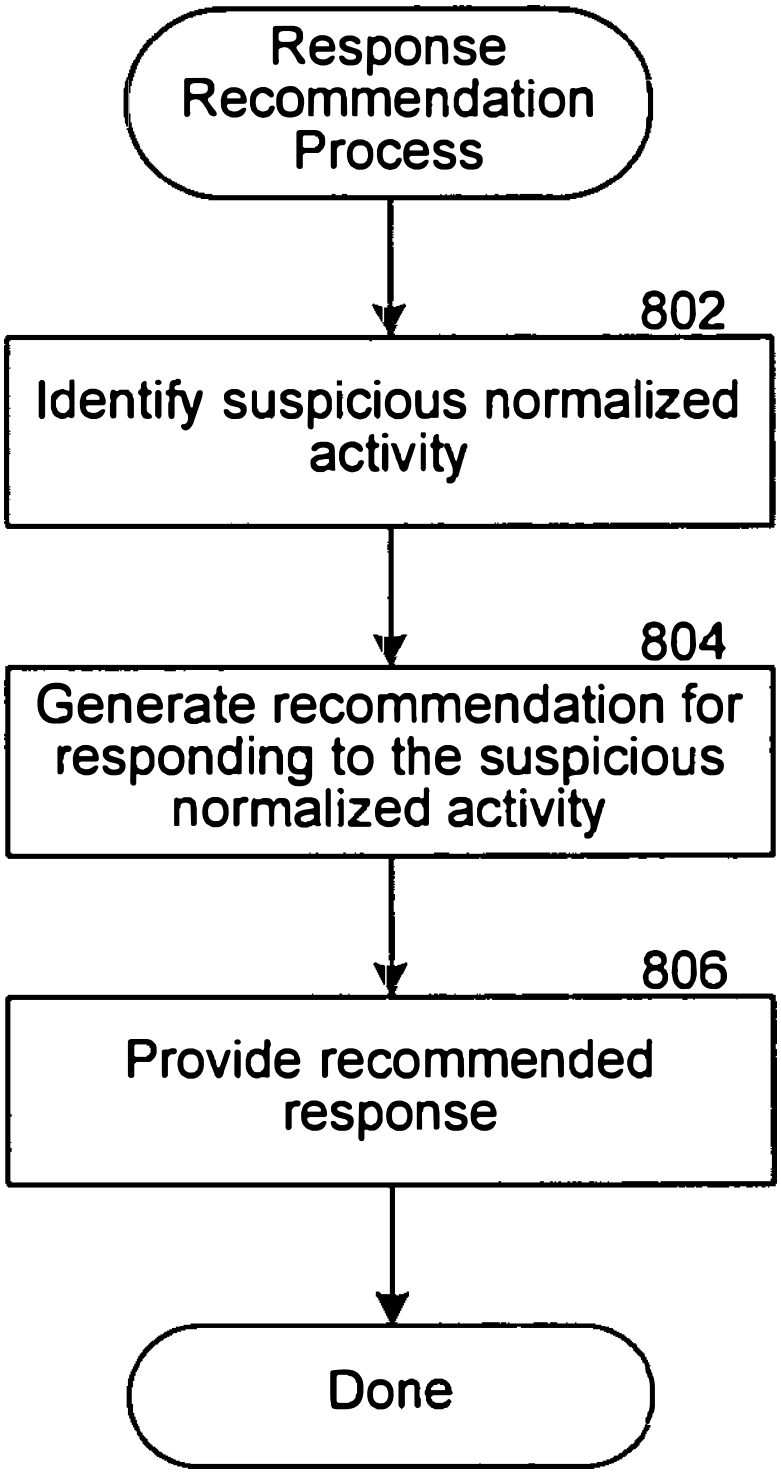

Response Recommendation Process
802
Identify suspicious normalized activity
804
Generate recommendation for responding to the suspicious normalized activity
806
Provide recommended response
Done

FIG. 8

MALICIOUS CODE INFECTION CAUSE-AND-EFFECT ANALYSIS

BACKGROUND

As dependence on computer systems and computer networks increases along with the sophistication and the frequency of attacks on computer systems and computer networks, the subject of computer security is becoming ever more prominent in the industry. Computer security techniques are presently used to protect users of the computer systems and the application programs and operating systems from malicious software ("malware")—e.g., viruses, worms, and Trojans—designed specifically damage or disrupt a computer system, and other undesirable activity.

To guard against the risk of malware, protection technologies, such as anti-malware software are often employed. For example, anti-malware software programs can scan computer systems to detect malware embedded within infected computer files. Once detected, the malware can be removed from infected files, the infected files may be quarantined, or the infected file may be deleted from the computer system.

Anti-malware software is currently designed to use known patterns, or signatures, to detect and remove malware infections, including "spyware" from computer operating systems and networks. For example, the anti-malware software can look for these patterns in an operating system's data stores, in memory, or over a variety of network protocols. After the failure or successful removal of the malware, anti-malware software typically provides the protected computer system administrator or network operator information surrounding the "cleaning event," which is typically referred to as an event or alert. For example, upon successfully or unsuccessfully cleaning malware off a computer system, the anti-malware provides a single event that declares that the machine is now "clean" and in a known good state. The information can include a friendly virus name, a list of the actions taken, and a date/time reference, amongst other information recorded at the time. A difficulty with the information provided after an infection is detected by today's anti-malware software is that system administrators are unable to determine the root cause or origin of that malicious code. Another difficulty with the information provided by the anti-malware software is that the information does not allow the system administrators to assess the scope of the damage caused subsequent to the initial infection.

SUMMARY

A malware analysis system for automating cause and effect analysis of malware infections is provided. The malware analysis system monitors and records computer system activities. Upon being informed of a suspected malware infection, the malware analysis system creates a time-bounded snapshot of the monitored activities that were conducted within a time frame prior to the notification of the suspected malware infection. The malware analysis system may also create a time-bounded snapshot of the monitored activities that are conducted within a time frame subsequent to the notification of the suspected malware infection. The malware analysis system provides the created snapshot or snapshots for further analysis. For example, the snapshots of the monitored activities may be analyzed to determine the cause and effect of the malware infection.

This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a high-level block diagram that illustrates selected components of a malware analysis system, according to some embodiments.

FIG. **2** is a flow diagram that illustrates the processing of the system activity monitor component of the malware analysis system, according to some embodiments.

FIG. **3** is a flow diagram that illustrates the processing of the categorization component of the malware analysis system, according to some embodiments.

FIG. **4** is a block diagram that illustrates an example malware state model, according to some embodiments.

FIG. **5** illustrates a table of example mappings of normalized actions to pre- and post-notification malware states.

FIG. **6** illustrates a table of example mappings of recorded activities to normalized actions and associated malware states.

FIG. **7** is a flow diagram that illustrates the processing of the commonality analysis component of the malware analysis system, according to some embodiments.

FIG. **8** is a flow diagram that illustrates the processing of the response recommendation generator of the malware analysis system, according to some embodiments.

DETAILED DESCRIPTION

A method and system for automating cause and effect analysis of malware infections, is provided. In some embodiments, the malware analysis system executes on a computer system or device, such as a desktop computer system, a server, firewall, proxy, domain controller, router, etc., and monitors for common operating system and network activities that lead to the creation, persistence and proliferation of malware. Upon being informed of the detection of a malware infection or suspected infection, for example, by locally executing anti-malware software, the malware analysis system provides a snapshot (or evidence) of the activities that were conducted within a specified time frame before the detected or suspected infection for analysis. For example, the malware analysis system can provide a snapshot of the last ten minutes of the monitored activities (e.g., local system activity, network activity, etc.) prior to the detection of the infection. This snapshot may be referred to as a "pre-infection snapshot." In some embodiments, the malware analysis system can provide the snapshot of the activities to a remote (e.g., centralized) data store. The snapshot of the activities can then be analyzed to determine the "root cause" or source of the infection (i.e., the activities that led to the infection), including, for example, how the infected computer system was discovered, how the infected computer system was accessed, what operating system objects were manipulated, and the like. In effect, the malware analysis system creates a "crash dump" for malware infections that can be analyzed to determine, for example, the root cause of the infection.

In some embodiments, the malware analysis system can provide a snapshot of the activities that were conducted within a specified time frame after the detected or suspected infection for analysis. This snapshot may be referred to as a "post-infection snapshot." For example, the malware analysis system can provide a snapshot of the monitored activities (e.g., local system activity, network activity, etc.) for the subsequent ten minutes after the detection of the infection.

The snapshot of the activities can then be analyzed to assess the damage and/or characterize any additional threats or damage to the infected computer system. For example, the snapshot of the subsequent activities can be analyzed to determine the activities that occurred subsequent to the infection, including identifying the operating system objects that were manipulated, additional communications that occurred with other computer systems, and the like. When applied across multiple computer systems and/or networks, a knowledgeable administrator, such as a security or system administrator, can analyze the snapshots (i.e., the snapshot of the activities prior to the detected/suspected infection and/or the snapshot of the activities subsequent to the detected/suspected infection) and use the result of the analysis to manually or automatically re-configure security policies in the environment to prevent future infections.

In some embodiments, the malware analysis system may monitor additional activities subsequent to the detection of an infection or suspected infection. For example, the malware analysis system may be configured to monitor a specific directory or directories in the file system when there are no infections or suspected infections on the computer system, and configured to monitor the entire file system when informed of an infection or suspected infection.

In some embodiments, the malware analysis system may apply a state model for malware to normalize and categorize the monitored activities in the pre-infection snapshot and/or the post-infection snapshot to aid in constructing cause and effect relationships. The state model may describe the phases of a malware infection within and about the computer system, such as propagation, activation, carrier, execution and persistence, and allows many seemingly normal system activities to be identified as part of a malicious code's lifespan when categorized in the context of a suspected infection. Applying the state model also allows for categorizing the stage in which the malware was in, and aids in determining the extent of an infection.

In some embodiments, the malware analysis system may perform commonality analysis on the normalized activities in the pre-infection snapshot and/or the post-infection snapshot to find any recurring activities. Once the malware analysis system discovers what the commonality is between the snapshots, the malware analysis system can tailor and provide a recommendation for responding to the detected infection. The malware analysis system can utilize an expert system to tailor and provide a recommendation based on the commonality analysis. For example, the commonality analysis process may indicate that each of four infected machines visited the same web site prior to being infected. Here, the malware analysis system may determine that this web site most likely served the malware to each of the infected machines. Using the expert system, the malware analysis system may recommend that this web site be "blocked."

FIG. **1** is a high-level block diagram that illustrates selected components of a malware analysis system **100**, according to some embodiments. The malware analysis system comprises a system activity monitor component **102**, a categorization component **104**, a commonality analysis component **106**, and a response recommendation generator **108**. The system activity monitor component provides runtime monitoring of the operating system resources for changes to the file system, common file formats, configurations (registry), network activities, use of common application program interfaces (APIs), or any other operating system object. The system activity monitor component may run on and monitor the activity of a computer system, such as, by way of example, a local desktop operating system, a server or network device. While executing, the system activity monitor component records the monitored activities in a data store, which may be in memory, on physical media, or other logical data store. The system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example: the identified operating system object or network object involved in the monitored activity (e.g., file name, socket, IP address, logical paths, etc.); the details of the change (e.g., create file, listen on network socket, etc.); if applicable, the details of the object prior to the change; the source(s) of the change (e.g., process id, user security context, logical storage identity from which data originated, logical storage identity where the change occurred, network sources such as uniform resource locator (URL) or internet protocol (IP) address, the API call used to make the change, etc.); a date and time stamp in which the event occurred. The system activity monitor component can be notified of a malware infection. For example, the system activity monitor component may provide an API through which an anti-malware system, or other well-known intrusion detection systems, can pass a notification of a confirmed or suspected malware infection. Upon receiving the notification, the system activity monitor component creates and provides a time-bounded snapshot of activities that occurred before and/or after the notification of infection to the other components of the malware analysis system. For example, the system activity monitor component may provide the snapshot of activities to a centralized data store that is accessible by the other components of the malware analysis system.

The categorization component processes the snapshot of the activities that are provided by the system activity monitor component. In some embodiments, the categorization component normalizes the snapshot data for use by the commonality analysis component. Normalization of the snapshot data may involve labeling and associating individual activities that are contained in the snapshots with predetermined, common categories of system activities. Common categories of system activities may include, for example, file transfer, web file transfer, instant messaging, remote authentication, remote control access, outbound network connection, application launch, file copy from media, file created, configuration change, etc. In some embodiments, the categorization component may map the normalized activities along a malware state model, with the initial infection notification as the reference for pre- and post-infection activities. The malware state model may comprise a multiple number of different malware states, and the categorization component may intelligently map each normalized activity in the snapshot to a malware state. Mapping the normalized activities to the malware states aids in determining the sequence of events that make up the infection. The mapping of the normalized activities and the malware states may be tuned (i.e., configured) by, for example, an administrator. The categorization component may provide the normalized snapshot data and/or the mapping of the normalized activities to the malware states to the other components of the malware analysis system.

The commonality analysis component compares the normalized snapshots and differentiates the normalized snapshots that are related to the same malware infection. In some embodiments, upon every comparison, the commonality analysis component, upon performing a comparison of the normalized snapshots, labels or tags the like-activities (i.e., duplicates) as "suspicious" with a given or specified malware state, and the unlike activities or events as "potentially normal." The activities that are tagged as potentially normal can be later filtered. After numerous comparisons, only certain reoccurring activities are likely to remain, each categorized as

either potentially normal or suspicious with a malware state. The commonality analysis component may make the results of the categorization available to administrators. For example, an administrator can then use this information to mark reoccurring activities as malicious (i.e., part of the infection). The commonality analysis component may also provide the results of its processing to the other components of the malware analysis system. In some embodiments, the commonality analysis component may call out to (i.e., obtain information from) other event sources, such as, by way of example, servers, firewalls, proxies, etc., to conduct further categorization and commonality analysis.

The response recommendation generator processes the information provided by the commonality analysis component and generates a recommendation for responding to the malware infection. The response recommendation generator may generate the recommendations or guidance based on the activities (e.g., the normalized activities) that are determined to be part of the detected malware infection. In some embodiments, the response recommendation generator may use an expert system that comprises a plurality of rules that are applied to determine a recommendation or multiple recommendations based on the results of the categorization provided by the commonality analysis component. The recommendations can be associated with specific types of system activities. For example, a recommendation may be to stop file transfers by implementing packet filtering or changes in system policy. The response recommendation generator may make the recommendations available to system or security administrators.

In a typical deployment of the malware analysis system, the various components of the malware analysis system are deployed over a multiple number of computer systems and devices. For example, the system activity monitor component may execute on all of the computer systems that are to be monitored, and one or more of the categorization component, the commonality analysis component, and the response recommendation generator may execute on one of the aforementioned computer systems or another, different computer system. One skilled in the art will appreciate that the malware analysis system may be deployed in other ways. For example, a remotely executing system activity monitor may remotely monitor the activities on certain types of computer systems, such as network devices. Moreover, one or more of the categorization component, the commonality analysis component, and the response recommendation generator may not be contained in the malware analysis system.

The computing device on which the malware analysis system is implemented may include a central processing unit, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and storage devices (e.g., disk drives). The memory and storage devices are computer-readable media that may contain instructions that implement the system. In addition, the data structures and message structures may be stored or transmitted via a data transmission medium, such as a signal on a communications link. Various communication links may be used, such as the Internet, a local area network, a wide area network, a point-to-point dial-up connection, a cell phone network, and so on.

Embodiments of the system may be implemented in various operating environments that include personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, programmable consumer electronics, digital cameras, network PCs, minicomputers, mainframe computers, network devices, distributed computing environments that include any of the above systems or devices, and so on. The computer systems may be cell phones, personal digital assistants, smart phones, personal computers, programmable consumer electronics, digital cameras, and so on.

The system may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, and so on that perform particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

FIG. **2** is a flow diagram that illustrates the processing of the system activity monitor component of the malware analysis system, according to some embodiments. By way of example, the system activity monitor component may execute as a runtime process that may use any of a variety of well-known monitoring techniques to monitor operating system and/or network activities. In block **202**, the system activity monitor component monitors predetermined activities on or about the computer system. For example, the predetermined activities to monitor may be specified by an administrator in a policy that contains settings for the system activity monitor component. In block **204**, the system activity monitor component records the monitored activities. In one embodiment, the system activity monitor component may record the activities in a sequential or circular data store in memory, on physical media, or other logical data store. In block **206**, the system activity monitor component determines whether it received notification of a suspected malware infection. If no notification is received, the system activity monitor component returns to block **202** and continues to monitor the predetermined activities. If a notification of a suspected malware infection is received, then, in block **208**, the system activity monitor component creates a pre-infection snapshot of the monitored activities. For example, the system activity monitor component may create a snapshot of the previous X seconds of the monitored activities that were recorded prior to receiving the notification event. The amount of monitored activities to include in the snapshot (e.g., the X seconds) may be specified by an administrator in the policy associated with the system activity monitor component. In block **210**, the system activity monitor component creates a post-infection snapshot of the monitored activities. For example, the system activity monitor component may create an additional snapshot of the subsequent Y seconds of recorded monitored activities after receiving the notification event. In some embodiments, the activities that are monitored subsequent to receiving the notification event may be different from the activities that are monitored prior to receiving the notification event. In block **212**, the system activity monitor component provides the created snapshots for further processing and/or analysis. For example, the system activity monitor component may provide the snapshots to a remote data store that is accessible by the other components of the malware analysis system and/or an administrator. The other components of the malware analysis system can process the pre-infection snapshot and the post-infection snapshot together when both are ready, or process each snapshot immediately and independent of each other.

One skilled in the art will appreciate that, for this and other processes and methods disclosed herein, the functions performed in the processes and methods may be implemented in differing order. Furthermore, the outlined steps are only exemplary, and some of the steps may be optional, combined with fewer steps, or expanded into additional steps.

FIG. **3** is a flow diagram that illustrates the processing of the categorization component of the malware analysis sys-

tem, according to some embodiments. In block **302**, the categorization component normalizes the activities contained in the snapshots provided by the system activity monitor component into predetermined set of common categories of system or network events (also referred to as "normalized actions"). In block **304**, the categorization component maps each normalized activity to a malware state as provided by a malware state model. FIG. **4** is a block diagram that illustrates an example malware state model, according to some embodiments. The malware state model describes how malware typically behaves in a computer system, and may be specified by an administrator. One skilled in the art will appreciate that the malware state model may be comprised of different states (e.g., the states in the model may change, be removed, or other states may be added) than the states that are depicted.

As depicted in FIG. **4**, the malware states comprise a propagation state **402**, an activation state **404**, a carrier state **406**, an execution state **408**, and a persistence state **410**. The malware infection targets are generated in the propagation state. For example, a computer system needs to be targeted in order for it to be infected by the malware. This may occur, for example, when a user uses the computer system to visit a source of the malware, such as a web site. Once the malware successfully propagates onto the computer system (i.e., finds a target) (stage **1**), it is considered to be in the activation state. When the malware on the computer system successfully executes its payload (stage **2**), the malware is considered to proceed from the activation state to the execution state. Otherwise, if the activation failed on the computer system (stage **3**), the malware is considered to move back to the propagation state. Sometimes, some other carrier or secondary channel is used to bring the additional code, instructions, data, etc. that are needed by the malware, for example, to execute on the computer system stage **4**). For example, the user of the computer system may be tricked into clicking on another web site link, which causes the downloading of the additional code, instructions or data. In another example, the malware, such as a worm, could pull from a remote data source additional code, instructions or data after the malware has been activated on the computer system. In these instances, the malware on the computer system is considered to be in the carrier state. Then, when the malware on the computer system successfully executes its payload (stage **5**), the malware is considered to proceed from the carrier state to the execution state. When the malware successfully installs itself on the computer system (stage **6**), the malware is considered to proceed from the execution state to the persistence state. In the persistence state, the malware may communicate with external sources (referred to as external communication) (stage **7**). For example, hackers can use the external communication (e.g., instant messaging channels, etc.) to send commands to the computer system to remotely control the compromised computer system. When the hackers, or some other controlling entity, remotely commands and controls the computer system (stage **8**), the malware is considered to proceed back to the execution state. When the malware gains durability on the computer system by, for example, persisting runtime activities, surviving operating system changes, etc. (stage **9**) the malware is considered to proceed from the execution state to the propagation state. If the malware does not persist itself on the computer system (stage **10**), the malware is considered to proceed from the execution state to the propagation state. Malware in the propagation state can propagate itself within the same computer system or to another computer system (i.e., another target). When an infection is reported, for example, by the anti-malware software (stage **11**), the malware is considered to be in the execution state. The system activity monitor component can use the report of the infection as the base time (e.g., t**0**) from which to create the time-bounded snapshots of recorded activities. Thus, t**0** can denote the time the malware executed on the computer system. In other embodiments, the malware may be considered to be in a state other than the execution state when an infection is reported. For example, a network module may detect a port scan, and indicate the initial state to be propagation.

FIG. **5** illustrates a table **500** of example mappings of normalized actions to pre- and post-notification malware states. The categorization component can use the table of mappings of normalized actions to intelligently map each normalized activity to a malware state. The table of example mappings of normalized actions comprises one or more mapping records **502**, and each mapping record is shown comprising, by way of example, four fields including a normalized action field **504**, a pre-notification malware state field **506**, a post-notification malware state field **508**, and a state transition field **510**. The normalized action field identifies a common category of system or network event. The activities that are recorded and contained in the pre-infection and post-infection snapshots are mormalized into one of these common categories. The pre-notification malware state field specifies a malware state to assign to the common category of system or network event that is identified in the corresponding normalized action field if the identified common category of system or network event occurred prior to the notification of the malware infection. Stated differently, if the normalized action is contained in the pre-infection snapshot, the normalized action is assigned the malware state that is specified in the corresponding pre-notification malware state field. The post-notification malware state field specifies a malware state to assign to the common category of system or network event that is identified in the corresponding normalized action field if the identified common category of system or network event occurred subsequent to the notification of the malware infection. Stated differently, if the normalized action is contained in the post-infection snapshot, the normalized action is assigned the malware state that is specified in the corresponding post-notification malware state field. The state transition field specifies or denotes the normalized activity that is recognized as a result of moving from one state to another state. The state transition may be considered a transition condition, and is recorded to aid comparative analysis. The mappings of the normalized actions to corresponding pre- and post-notification malware states are configurable. For example, an administrator can tune the mappings based on past performance/accuracy.

FIG. **6** illustrates a table **600** of example mappings of recorded activities to normalized actions and associated malware states. The table of example mappings of recorded activities comprises one or more mapping records **602**, and each mapping record is shown comprising, by way of example, four fields including a time field **604**, a recorded activity field **606**, a normalized action field **608**, and an associated malware state field **610**. The time field specifies, relative to "t**0**" (i.e., the time at which the notification of the malware infection is received by the malware analysis system), the time the activity identified in the corresponding recorded activity field was recorded by the system activity monitor component of the malware analysis system. The normalized action field identifies a normalized action that maps to the activity that is identified in the corresponding recorded activity field. The associated malware state field identifies the malware state that is assigned to the normalized action that is identified in the corresponding normalized

action field, based on the time the activity that corresponds to the normalized action was recorded.

By way of example, the record for the time "t0−1" indicates that the recorded activity is "Outbound UDP connection on port **69** (TFTP)." The categorization component may have normalized this recorded activity to be "File Transfer" as indicated in the normalized action field. Using the table of example mappings of normalized actions (FIG. **5**), the categorization component assigns this normalized action to the "Carrier" malware state since this normalized action occurred prior to the notification of the malware infection. Assuming that this normalized action occurred subsequent to the notification of the malware infection, the categorization component would assign this normalized action to the "persistence" malware state. In a similar manner, the categorization component can use (1) a predetermined mapping of recorded activities to corresponding normalized actions, and (2) a predetermined mapping of the normalized actions to corresponding pre-notification malware states and post-notification malware states (e.g., the example table illustrated in FIG. **5**) to normalized each recorded activity contained in the snapshots and assign to each normalized activity (i.e., normalized action) an appropriate malware state based on the time the activity was recorded.

Referring again to FIG. **3**, in block **306**, the categorization component provides the mapped normalized activities for further processing and/or analysis.

FIG. **7** is a flow diagram that illustrates the processing of the commonality analysis component of the malware analysis system, according to some embodiments. In block **702**, the commonality analysis component compares the normalized activities that are recorded for a particular malware infection to previously stored normalized activities for the same malware infection. For example, the previously stored normalized activities may be the normalized activities which were obtained from a snapshot provided by a different computer system that was similarly infected. For each comparison (block **704**), the commonality analysis component performs blocks **706** to **710**, until all the comparisons are performed (block **712**). In block **706**, the commonality analysis component tags similar activities as being suspicious. In block **708**, the commonality analysis component assigns a malware state to the suspicious activity. For example, the commonality analysis component may determine an appropriate malware state from the table of example mappings of normalized actions to pre- and post-notification malware states (FIG. **5**). In block **708**, the commonality analysis component tags unlike (i.e., dissimilar) activities as being potentially normal. For example, these potentially normal activities are the activities that were recorded in only one or a few of the similarly infected computer systems, and are likely not associated with the malware infection. Subsequent to performing all the comparisons of the normalized activities that were recorded and provided in the pre-infection and/or post-infection snapshots from the infected computer systems for the particular malware infection (blocks **704** to **712**), the commonality analysis component, in block **714**, filters out the potentially normal activities (i.e., the normalized activities which were tagged as being potentially normal). In block **716**, the commonality analysis component provides the results for further processing and/or analysis. For example, the results include the normalized activities which were tagged as being suspicious, but do not include the normalized activities which were tagged as being potentially normal and, thus, filtered by the commonality analysis component. One skilled in the art will appreciate that the commonality analysis component may use any of a variety of well-known comparative analysis techniques to identify statistically significant commonalities.

FIG. **8** is a flow diagram that illustrates the processing of the response recommendation generator of the malware analysis system, according to some embodiments. In block **802**, the response recommendation generator identifies a suspicious normalized activity. This may be a normalized activity that was tagged as being suspicious by the commonality analysis component. In block **804**, the response recommendation generator generates a recommendation or multiple recommendations for responding to the suspicious normalized activity. In some embodiments, the response recommendation generator can utilize an expert system to generate the appropriate recommended response. In block **806**, the response recommendation generator provides the recommended response. For example, the response recommendation generator can provide the recommended response to an administrator.

Example Scenarios

In a first example, ten computer system users visit a web site using their personal computers. Each of the computer systems are running anti-malware software and the malware analysis system, which is monitoring all activities that modify local data stores (e.g., file system, registry, databases, etc.) and network activities (e.g., TCP/IP connections, web site visits, etc.). Each of the ten computer systems become infected with a known spyware application, served by the web site, which has mild variations from previously known versions to evade or complicate removal by anti-malware software. Upon detecting the variant (i.e., the spyware application), locally installed anti-malware software on each computer system attempts to clean the infection, and creates an "event" describing the actions it took and an indication of what succeeded or failed. The malware analysis system running on each of the computer systems receives an indication of the detection of the variant, and provides a snapshot of the last five minutes and subsequent five minutes of local system and network activity to a centralized data store. The malware analysis system then normalizes the snapshot data on the centralized data store, and conducts commonality analysis on the ten provided snapshots to find any recurring activities. The malware analysis system performs the commonality analysis and discovers that each of these ten computer systems visited the same web site within one minute prior to the infection. The malware analysis system may alert the system operator of all ten infected computer systems to the pattern, and provide a recommendation to block this web site at their organization's egress points to prevent future infections.

In another example, ten computer system users visit a web site using their personal computers. Each of the computer systems are running anti-malware software and the malware analysis system, which is monitoring all activities that modify local data stores (e.g., file system, registry, databases, etc.) and network activities (e.g., TCP/IP connections, web site visits, etc.). Each of the ten computer systems become infected with a known spyware application, served by the web site, which has mild variations from previously known versions to evade or complicate removal by anti-malware software. Upon detecting the variant (i.e., the spyware application), locally installed anti-malware software on each computer system attempts to clean the infection, and creates an "event" describing the actions it took and an indication of what succeeded or failed. The malware analysis system running on each of the computer systems receives an indication of the detection of the variant, and provides a snapshot of the

last five minutes and subsequent five minutes of local system and network activity to a centralized data store. The malware analysis system then normalizes the snapshot data on the centralized data store, and conducts commonality analysis on the ten provided snapshots to find any recurring activities. The malware analysis system performs the commonality analysis and discovers that each of these ten computer systems visited the same web site within one minute prior to the infection. The malware analysis system may request additional information regarding monitored activities during the same time period from other security event sources, such as, by way of example, egress firewalls, and receive the additional snapshots from each of these sources. The malware analysis system can then perform further commonality analysis on the newly provided snapshots. For example, the malware analysis system may discover additional computer systems that have visited the same web site, as well as several other computer systems that have been contacted by the web site in return. These new computer systems may or may not be running anti-malware software or the malware analysis system. The malware analysis system may alert the system operator of all ten infected computer systems to the pattern, and provide a recommendation to block this web site at their organization's egress points to prevent future infections. In addition, malware analysis system may alert the system operator of the several other system systems that may be infected with the spyware or other malware, since these other computer systems also visited or were contacted by the known "bad" web site.

Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

We claim:

1. A method in a computer system for performing cause and effect analysis of a malware infection, the method comprising:

monitoring a plurality of activities on a first computer system;

recording the monitored activities;

upon receiving a notification of a suspected malware infection, creating a first pre-infection snapshot by storing an indication of recorded monitored activities as the first pre-infection snapshot, the first pre-infection snapshot indicating the monitored activities that were conducted within a time frame prior to the notification of the suspected malware infection;

receiving second pre-infection snapshots from second computer systems, the second pre-infection snapshot for a second computer system being created by storing an indication of recorded monitored activities of the second computer system that were conducted within a time frame prior to that second computer system receiving a notification of the suspected malware infection;

analyzing the first pre-infection snapshot by comparing the monitored activities of the first pre-infection snapshot to the monitored activities of second pre-infection snapshots of second computer systems to identify common activities represented by the first pre-infection snapshot and the second pre-infection snapshots that are similar across the first pre-infection snapshot and the second pre-infection snapshots; and

generating a recommendation for processing the suspected malware infection based on the identified common activities,

wherein the first computer system and the second computer systems are on different machines.

2. The method of claim **1** further comprising:

upon receiving the notification of the suspected malware infection, creating a post-infection snapshot, the post-infection snapshot containing the monitored activities that are conducted within a time frame subsequent to the notification of the suspected malware infection; and

providing the post-infection snapshot for analysis.

3. The method of claim **2**, wherein the post-infection snapshot contains at least one monitored activity that is not contained in the pre-infection snapshot.

4. The method of claim **1**, wherein the activities include at least one local computer system activity.

5. The method of claim **1**, wherein the activities include at least one network activity.

6. The method of claim **1** further comprising normalizing the activities contained in the pre-infection snapshot into corresponding normalized actions.

7. The method of claim **6** further comprising mapping each normalized activity to a corresponding malware state.

8. A method in a computer system for performing cause and effect analysis of a malware infection, the method comprising:

providing a malware state model indicating how malware behaves on an infected computer system, the malware state model having malware states and indications of activities for transitioning from one malware state to another malware state;

providing a mapping of normalized activities to malware states;

receiving a pre-infection snapshot from a plurality of machines, each of the pre-infection snapshots identifying monitored activities that were conducted within a time frame prior to a suspected malware infection on each machine of the plurality of machines;

for each received pre-infection snapshot, normalizing the activities of the pre-infection snapshot by assigning a predetermined category to each monitored activity of the pre-infection snapshot; and

comparing the normalized activities of the pre-infection snapshots to the normalized activities of other pre-infection snapshots to identify normalized activities that are similar, normalized activities being similar when their assigned categories are the same;

when the comparison indicates the normalized activities are similar, tagging similar normalized activities as being suspicious; and

assigning a malware state to the tagged normalized activities based on the provided malware state model and the providing mapping of normalized activities to malware states;

when the comparison indicates that normalized activities are not similar, not tagging the normalized activities as being suspicious; and

generating a recommendation for processing the infection based on the normalized activities that have been tagged as suspicious and the assigned malware states of the normalized activities that have been tagged as suspicious.

9. The method of claim **8** further comprising assigning a malware state to each activity tagged as being suspicious.

10. The method of claim **8** further comprising:

receiving a post-infection snapshot from the plurality of machines, each of the post-infection snapshots containing monitored activities that are conducted within a time frame subsequent to the suspected malware infection on each machine of the plurality of machines; and

comparing the monitored activities to each other and tagging similar activities as being suspicious.

11. The method of claim **10** further comprising assigning a malware state to each activity tagged as being suspicious.

12. The method of claim **10** further comprising comparing the monitored activities in the post-infection snapshots to each other and tagging unlike activities as being potentially normal.

13. The method of claim **8** further comprising providing the activities tagged as being suspicious for analysis.

14. The method of claim **10**, wherein the recommendation for responding to the at least one activity tagged as being suspicious is generated using an expert system.

15. A computer system for performing analysis of a malware infection to determine a root cause of the malware infection, comprising:

a memory storing computer-executable instructions of

a component that receives pre-infection snapshots from a plurality of machines suspected of

being infected with malware, each of the pre-infection snapshots identifying monitored activities that were conducted at each machine suspected of the plurality of machines suspected of being infected

with malware within a time frame prior to the machine being suspected of being infected with malware;

a component that compares the monitored activities of the pre-infection snapshot to identify monitored activities that are common across multiple pre-infection snapshots;

a component that tags as being suspicious the monitored activities that are common across multiple pre-infection snapshots; and

a component that provides an indication of the root cause of the malware infection for an infected machine based on the monitored activities that are tagged as being suspicious; and

a processor that executes the computer-executable instructions stored in the memory.

16. The computer system of claim **15** wherein the component that compares normalizes the monitored activities before identifying monitored activities that are common.

17. The computer system of claim **16** wherein the component that tags assigns a malware state to each activity tagged as being suspicious.

18. The computer system of claim **15** wherein the component that tags assigns a malware state to each activity tagged as being suspicious.

19. The computer system of claim **15** including:

a component that receives post-infection snapshots from the plurality of machines suspected of being infected with malware, each of the post-infection snapshots containing monitored activities that were conducted within a time frame after the suspected infection on the infected machine; and

a component that compares and tags as being suspicious the monitored activities of the post-infection snapshots.

20. The method of claim **1** wherein the analyzing includes detecting that the snapshots indicate that the first computer system and the second computer system visited the same web site and wherein the generated of the recommendation indicates to block that web site.

* * * * *