

REAL-TIME REPRESENTATION OF SECURITY- RELEVANT SYSTEM STATE

RELATED APPLICATIONS

5 **[0001]** This patent application claims priority to U.S. Utility patent application entitled “Real-Time Representation of Security-Relevant System State” with Serial No. 13/728,746 filed December 27, 2012, which is fully incorporated herein by reference.

BACKGROUND

10

[0002] With Internet use forming an ever greater part of day to day life, security exploits that steal or destroy system resources, data, and private information are an increasing problem. Governments and businesses devote significant resources to preventing intrusions and thefts related to these security
15 exploits. Security exploits come in many forms, such as computer viruses, worms, trojan horses, spyware, keystroke loggers, adware, and rootkits. These exploits are delivered in or through a number of mechanisms, such as spearfish emails, clickable links, documents, executables, or archives. Some of the threats posed by security exploits are of such significance that they are
20 described as cyber terrorism or industrial espionage.

[0003] To meet the threats posed by these security exploits, a number of tools capable of retrospective analysis of system performance and state have been developed. For example, the BackTracker tool described in “Backtracking Intrusions” by Samuel T. King and Peter M. Chen (ACM

SIGOPS Operating Systems Review – SOSP '03, Volume 37, Issue 5,

December 2003, pgs. 223-236) automatically identifies potential sequences of steps that occurred in an intrusion by analyzing a comprehensive log of system activities and data. While such tools can detect security exploits and their
5 manners of operation, they can only operate retrospectively and thus place those attacked at a disadvantage, always one step behind the attacker. The reason these tools can only operate retrospectively is the enormous computational cost of building the comprehensive log of system activities and data that is needed by the tools. In fact, the computational cost of writing
10 events to the log is often higher than the cost of the events themselves. Each event may be written multiple times and writes to the logs often must cross input/output boundaries.

BRIEF DESCRIPTION OF THE DRAWINGS

15 [0004] The detailed description is set forth with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items or features.

20 [0005] FIG. 1 illustrates an example situational model for representing in substantially real-time a security-relevant state of a monitored device and agent modules for building the situational model and for utilizing the situational model to validate state information.

[0006] FIG. 2 illustrates example networks connecting monitored device(s) of a client network to a remote security service and to an intermediate device of the client network, each of the remote security service, the intermediate device, and the monitored device(s) being capable of building and maintaining a situational model.

[0007] FIG. 3 illustrates a component level view of a computing device capable of acting as a security service device, a monitored device, or an intermediate device.

[0008] FIG. 4 illustrates an example process for validating, by a monitored device, state information based on security-relevant information stored by a situational model.

[0009] FIG. 5 illustrates an example process for building, by a remote security service, a situational model of security-relevant state for a group of monitored devices.

DETAILED DESCRIPTION

Overview

[0010] This disclosure describes, in part, a situational model representing security-relevant state of a monitored device. The situational model is constructed with security-relevant information in substantially real-time as execution activities of the monitored device associated with the security-relevant information are observed. The security-relevant information includes information in the scope of a configuration of the situational model that

represents a change of state. The represented security-relevant state may include a current state and a past state of the monitored device. Also, the situational model may be used to validate state information associated with execution activities. Both the constructing of the situational model and the querying of the situational model to validate state information may be performed by one or more modules of the monitored device, the modules and situational model comprising components of a kernel-level security agent. Further, the situational model may serve as a cache of information received from a remote security service, of information that is computationally expensive to generate or fetch, or of information that is high-latency to fetch.

[0011] In various embodiments, a remote security service may configure the monitored device, including configuring the situational model and the one or more modules, and may build an additional situational model representing security-relevant state of a group of monitored devices. The remote security service may build the additional situational model using security-relevant information received from the monitored devices. The monitored devices may provide the security-relevant information responsive to the configuration received from the remote security service. Further, in some embodiments, the configuring may be performed by the remote security service and the building of the additional situational model may be performed by an intermediate device of a client network that includes at least a part of the monitored devices. In such embodiments, the additional situation model may represent a state of the monitored devices of the client network.

[0012] In further embodiments, the remote security service or the kernel-level security agent may utilize information stored in the situational model but no longer present on a host operating system of a monitored device. This information may be about earlier processes or files in a chain of execution that
5 led to an observable behavior of the monitored device. For example, a maliciously crafted document crashes a legitimate application and runs attack code in its place. The attack code launches a script which launches a new copy of Word, thus giving the appearance that the legitimate application did not crash. This historical information may be used by the remote security service
10 or kernel-level security agent, for instance, in further analysis or in determining a responsive action.

[0013] FIG. 1 illustrates an example situational model for representing in substantially real-time a security-relevant state of a monitored device and agent modules for building the situational model and for utilizing the situational
15 model to validate state information. As illustrated, the monitored device 102 may include one or more processors 104 and memory 106 storing agent module(s) 108 and a situational model 110. The agent module(s) 108 may receive notifications of events 112 included in or generated by execution activities 114 of the processor(s) 104. For at least some types of events 112,
20 the agent module(s) 108 may retrieve state information 116 associated with the execution activities 114. The agent module(s) 108 may then utilize the events 112 and/or state information 116 to build the situational model 110, providing those events 112 and/or state information 116 to the situational model 110 as

security-relevant information 118. The situational model 110 may include a receiver 120 to receive the security-relevant information 118 and store it in a database 122 of the situational model 110. In this manner, the agent module(s) 108 and situational model 110 may build the situational model 110 into a representation of current and past system states. In some embodiments, the agent module(s) 108 may also query 124 the situational model 110 using a query interface 126 of the situational model 110, the query interface 126 enabling retrieval of security-relevant information 128. The agent module(s) 108 may then utilize the security-relevant information 128 to validate additional state information 116. Also, to control the size of the situational model 110 while retaining important security-relevant information 118, the situational model 110 may include a pruning component 130 to selectively delete security-relevant information 118 based on configuration.

[0014] In various embodiments, the monitored device 102 may be a server or server farm, multiple, distributed server farms, a mainframe, a work station, a personal computer (PC), a laptop computer, a tablet computer, a personal digital assistant (PDA), a cellular phone, a media center, an embedded system, or any other sort of device or devices. When implemented on multiple computing devices, the monitored device 102 may distribute the agent module(s) 108 and situational model 110 among the multiple computing devices. In some implementations, the monitored device 102 represents one or more virtual machines implemented on one or more computing devices. An

example computing device capable of serving as a monitored device 102 is illustrated in FIG. 3 and described below with reference to that figure.

[0015] Also, in some embodiments, the monitored device 102 may belong to a client or client network that subscribes to a security service provider for security services. The monitored device 102 may thus be connected to the remote security service via a network. Through such a network, the monitored device 102 may receive an initial installation of agent module(s) 108 and situational model 110 and may receive configuration updates at various times. The monitored device 102 may also provide security-relevant information 118 and notifications to the remote security service based on triggers included in the configuration received from the remote security service.

[0016] In various embodiments, the processor(s) 104 is a central processing unit (CPU), a chipset component, a graphics processing unit (GPU), another processing unit or component known in the art, or any combination of processing units. The execution activities 114 of the processor(s) 104 may be any active processes, mutexes, etc. currently being executed by the processor(s) 104.

[0017] In various embodiments memory 106 is volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. The memory 106 may further be any tangible removable or non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. Examples of memory 106 include, but are not limited

to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other tangible medium which can be used to store the desired
5 information and which can be accessed by the monitored device 102.

[0018] In various embodiments, the agent module(s) 108 may be one or more modules of a kernel-level security agent. Such a kernel-level security agent may include event consumers that receive notifications of events 112 associated with execution activities 114, filters, an event bus that route events
10 112 to other ones of agent module(s) 108, correlators that track types of events 112, actors that gather state information 116 and act upon events 112, and a situational model 110. The kernel-level security agent may be installed by and configurable by a remote security service, receiving, and applying while live, reconfigurations of agent module(s) 108 and situational model 110. An
15 example kernel-level security agent is described in greater detail in U.S. patent application serial number 13/492,672, entitled "Kernel-Level Security Agent" and filed on June 8, 2012.

[0019] In some embodiments, the agent module(s) 108 include event consumers, and the event consumers may be configured to listen for
20 notifications of events 112. The types of events 112 listened for may be specified by configuration received from a remote security service. The event consumers may then provide the events 112 to an event bus through one or more filters, the filters selectively relaying the events 112 from the event

consumers to the event bus based on the received configuration. Filters and the event bus may then provide the events 112 one or more of a correlator, an actor, or to the situational model 110. The filters and event bus may instead or additionally provide the events 112 to a remote security service. Besides
5 receiving events 112 from the event consumers, the filters and event bus may receive configurations and security-relevant information 118 from the remote security service. This information/configuration may be provided to other agent module(s) 108 (e.g., actors) and/or may be cached in the situational model 110.

10 **[0020]** Correlators may track specific types of events whose occurrence, e.g., in a certain aggregate, is specified to be security-relevant information 118 by the received configuration. Once the correlators determine that one or more received events 112 constitute security-relevant information 118, the correlators provide the security-relevant information 118 to the receiver 120 of
15 the situational model 110.

[0021] Actors may participate in building the situational model 110, in utilizing the situational model 110 to validate state information 116, and in utilizing the situational model 110 as a cache. Upon receiving events 112 from filters and the event bus, the actors may determine, based on the configuration
20 received from the remote security service and on the type of the event whether to retrieve state information 116 from the execution activities 114 that include or generated the event 112. Such state information 116 may comprise a copy of a process, information drawn from a memory of the process, contents of a

call stack associated with that process, data written or read by that process, etc.

The actors may then provide one or both of the events 112 and state information 116 to the receiver 120 of the situational model 110.

[0022] In some embodiments, in addition to or in place of providing state
5 information 116 to the situational model 110, the actors may query 124 the
query interface 126 of the situational model 110 for query results, such as
security-relevant information 128, and may utilize the query results to validate
the state information 116. Validation may simply comprise a comparison of
the state information 116 to the security-relevant information 128 (which may
10 comprise state information from a prior point or points in time) to determine if
there is a difference. In other embodiments, validation may comprise a more
complex analysis of query results and state information 116. In response to a
negative result (e.g., the state information 116 not matching the security-
relevant information 128), the actors generate a validation error. Responsive to
15 a validation error, the actors may respond in a number of ways. For example,
the actors may perform at least one of reporting the event 112 and/or state
information 116 associated with the validation error to a remote security
service, alerting a user of the monitored device 102, taking an action to
heal/remediate (e.g., halting the process associated with the validation error), or
20 noting the validation error in the situational model 110 and performing further
monitoring.

[0023] In further embodiments, actors may receive configurations,
externally derived metadata about object represented in the situation model

110, or other information from a remote security service and may provide those configurations, metadata, or information to the situational model 110 through the receiver 120, the situational model 110 serving as a cache of the configurations or information. Additionally, the actors may provide other
5 information that is computationally expensive to generate or fetch, or that is high-latency to fetch, to the situational model 110. Such information may be identified by the configuration of the agent modules 108 and/or situational model 110.

[0024] In various embodiments, the situational model 110 represents current
10 and past states of the monitored device 102. The past states maintained include at least a subset of past states, such as states that enhance forensic and policy understanding of the current state. As described above, the situational model 110 may have at least three roles. In a first role, the situational model 110 may serve as a historian, providing access to past states of the monitored device 102
15 that are no longer stored elsewhere on the monitored device 102. In a second role, the situational model 110 may service as a validator, maintaining an independent model of the state of the monitored device 102 that can be used to detect malicious modifications of the host operating system state storage of the monitored device 102. Thus, even when the act of modification is not detected,
20 its effects may be through comparison of current state to the independently maintained state in the situational model 110. In a third role, the situational model 110 provides a cache of configuration, information, and state that are

received from a remote security service, that are computationally expensive to generate or fetch, or that are high-latency to fetch.

[0025] In some embodiments, the receiver 120 receives security-relevant information 118, such as events 112 and/or state information 116 from filters, correlators, or actors of the agent modules 108 and updates the database 122 with the received security-relevant information 118. To receive the security-relevant information 118, the receiver 120 may implement an event receiver interface on the above-mentioned event bus of the agent modules 108. In some embodiments the receiver 120 may inform the agent modules 108 of an inconsistency between received security-relevant information 118 and security-relevant information 118 stored in the database 122. The receiver 120 may also generate an event to provide to a remote security service, such as initializing a backup of the database 122. Further, the receiver 120 may generate any number of events based on received security-relevant information 118 and on security-relevant information 118 stored in the database 122. Such events may be provided to a remote security service or used to drive local actions.

[0026] In various embodiments, the database 122 may be any sort of database, file, or collection of data capable of storing information. The database 122 is a store with a configurable set of collections, such as tables. Each collection may represent a type of object about which the database 122 stores data (i.e., in the form of security-relevant information 118) and may be indexed by an identifier for that object class. Each collection for a type of object may include an index field, modeled data attributes (e.g., security-

relevant information 118), and metadata defining a trust period for the collection. Each collection may also include one or more sub-collections, such as a sub-collection listing the loaded modules of a process. Such metadata may include the last “epoch” in which the collection data was written, and a mask (e.g., a bitmask) of attributes written in that epoch. As used herein, the term “epoch” may refer to a period of time in which all data acquired about a collection has consistent properties of liveness and trustworthiness. For ephemeral objects, epochs occur at system boot, when the kernel-level security agent restarts, or following rundown on an initial installation of a kernel-level agent. For persistent objects, an epoch occurs whenever mount of a backing store associated with the persistent objects is lost. Collections are also associated with a trust period, which defines the most recent epoch for which the collection can be trusted as an accurate representation. If the last epoch matches a current epoch, then all collection attributes set in the bitmask are within the trust period, and other collection attributes not set in the bitmask may be outside of the trust period. For example, the trust period for ephemeral objects is the current epoch. For persistent objects, the trust period may be the current epoch or a past epoch.

[0027] The types of objects about which the database 122 stores security-relevant information 118 may be specified by a configurable schema, the configurable schema and configuration updates to the configurable schema being received from the remote security service. For each object type, the schema may also include a flag indicating whether a collection is persistent or

ephemeral. An ephemeral object may be any object that exists only in system memory (e.g., random access memory). Such an ephemeral object will not itself survive a reboot, although the corresponding representation of the object in the database 122 may survive the reboot, depending on configuration.

5 Examples of ephemeral objects include processes, threads, etc. Persistent objects are any objects which have a backing store (e.g., are stored in permanent storage). Examples of persistent objects include files and the BIOS.

[0028] In various embodiments, the query interface 126 may be an interface used by filters, actors, correlators, other agent modules 108, or the remote
10 security service to access the database 122 via queries 124. The query interface 126 may support two types of queries 124. The first type of query 124 may be a query requesting only security-relevant information 128 that is within the trust period. The second type of query 124 may be for all security-relevant information 128. In response to the query 124, the query interface 126 may
15 return query results with flags indicating whether or not all of the returned security relevant information 128 is within the trust period and whether or not the trust period reaches back to a specific event, such as a reboot or a mounting of a file system. In further embodiments, the query interface 126 may support concurrent reading from and writing to the database 122.

20 [0029] In some embodiments, the pruning component 130 maintains the database 122, including pruning security-relevant information 118 from the database 122. The situational model 110 may include the pruning component 130 to ensure that the database 122 does not exceed a threshold size limit. The

threshold size limit may be determined, for example, by the configuration received from the remote security service. The pruning component 130 may be configured to prune any security-relevant information 118 that is no longer needed to satisfy a query 124. The pruning component 130 may also assign
5 different classifications and degrees of importance to different queries 124. This enables pruning of security-relevant information 118 needed to answer a low priority query when pruning of security-relevant information 118 no longer needed for any query 124 fails to sufficiently reduce the size of the database 122.

10 **[0030]** In pruning security-relevant information 118, the pruning component 130 may prune by collection, by record, or by attribute. In some embodiments, the pruning component 130 may use collection-based pruning for ongoing maintenance of the database 122 and attribute-based pruning responsive to high storage conditions. Collection-based pruning may involve pruning any
15 collections representing ephemeral objects from previous epochs, as those ephemeral objects are no longer on the system. The collection-based pruning may, however, retain collections representing ephemeral objects when those ephemeral objects are referenced by persistent objects.

20 **Example Network**

[0031] FIG. 2 illustrates example networks connecting monitored device(s) of a client network to a remote security server and to an intermediate device of the client network, each of the remote security service, the intermediate device,

and the monitored device(s) being capable of building and maintaining a situational model. As illustrated in FIG. 2, a network 202 connects monitored device(s) 102 to a remote security service 204. The remote security service 204 may include a situational model 206 representing states of the monitored device(s) 102 and security modules 208. In some embodiments, the monitored device(s) 102 may be connected to the network 202 through a client network 210, the client network 210 being associated with a client subscribing to the services of the remote security service 204. The client network 210 may also include an intermediate device 212 that maintains a situational model 214 specific to the client.

[0032] In some embodiments, the network 202 and client network 210 may each include any one or more networks, such as wired networks, wireless networks, and combinations of wired and wireless networks. Further, the network 202 and client network 210 may include any one or combination of multiple different types of public or private networks (e.g., cable networks, the Internet, wireless networks, etc.). For example, the network 202 may be a public network and the client network 210 may each be a private network. In some instances, computing devices communicate over the network 202 and client network 210 using a secure protocol (e.g., https) and/or any other protocol or set of protocols, such as the transmission control protocol/Internet protocol (TCP/IP).

[0033] In various embodiments, the devices implementing the remote security service 204 and the intermediate device 212 may each be or include a

server or server farm, multiple, distributed server farms, a mainframe, a work station, a personal computer (PC), a laptop computer, a tablet computer, a personal digital assistant (PDA), a cellular phone, a media center, an embedded system, or any other sort of device or devices. In one implementation, the
5 devices implementing the remote security service 204 represent a plurality of computing devices working in communication, such as a cloud computing network of nodes. When implemented on multiple computing devices, the remote security service 204 may distribute the modules and data 206-208 of the remote security service 204 among the multiple computing devices. In some
10 implementations, one or more of the devices implementing the remote security service 204 and the intermediate device 212 represent one or more virtual machines implemented on one or more computing devices. An example computing device capable of serving as a device implementing the remote security service 204 or the intermediate device 212 is illustrated in FIG. 3 and
15 described below with reference to that figure.

[0034] In various embodiments, the remote security service 204 may be a provider of information security services to individual users and client entities, such as maintenance and configuration of the agent modules 108 and situational model 110, threat modeling, and/or remediation. The security
20 modules 208 of the remote security service 204 may be configured to perform such information security services.

[0035] In further embodiments, the remote security service 204 may build and maintain a situational model 206 representing states of a group of

monitored devices 102. Such states may include current states, past states, or both. The remote security service 204 may receive security-relevant information 118 used to build the situational model 206 from the monitored devices 102 responsive to a configuration of those monitored devices 102.

5 Such security-relevant information 118 may be received in substantially real time as the execution activities 114 associated with the security-relevant information 118 are observed. The security modules 208 may configure the monitored devices 102, specifying what security-relevant information 118 the monitored devices 102 are to provide to the remote security service 204. Such
10 security-relevant information 118 may be information that, in isolation on a single monitored device 102 may or may not be meaningful, but when received from multiple monitored devices 102 is meaningful. For instance, if a single monitored device 102 is connected to a third party system, then that connection may or may not be meaningful. However, if twelve monitored devices 102 are
15 connected to that third party system, that aggregate of connections may indicate malicious activity.

[0036] The situational model 206 may include analogs for some or all of the above-described receiver 120, the database 122, the query interface 126, and the pruning module 130, or may include variants thereof. Thus, the description
20 of the situational model 110 and its subcomponents may be partly or entirely applicable to the situational model 206. Upon receiving the security-relevant information 118, the security modules 206 or other logic of the remote security

service 204 may store the security-relevant information 118 in the situational model 206.

[0037] In addition to storing security-relevant information 118, the situational model 206 may also support validation activities and security decisions. To support these activities and decisions, the situational model 206 may be queried. Such queries may be made by the security modules 208, by user-operators of the remote security service 204, or by other logic of the remote security service 204. The queries may be received by a query interface of the situational model 206, which may be an analog or variant of the query interface 126. The query interface of the situational model 206 may retrieve and return the sought security-relevant information 118. Those query results may then be used in validating other security-relevant information 118 received from the monitored devices 206 or in making security decisions. Such security decisions may include determining whether to generate a configuration update for one or more of the monitored devices 102, whether to alert one or more of the monitored devices 102 of malicious activity, whether to continue monitoring the one or more monitored devices 102 before taking further action, or whether to take a remediation action, such as halting a process. The security modules 208 may then act upon such a security decision.

[0038] In further embodiments, the situational model 206 may serve as a backup of situational models 110 of monitored devices 102. The remote security service 204 may receive at least a part of a situational model 110 periodically and may store that part in the situational model 206. Such

delayed, periodic backup may avoid the overhead of using the situational model 206 as a real-time backup of the situational models 110.

[0039] In some embodiments, the remote security service 204 may also provide social aspects to the security services, forming groups of those users and/or client entities and automatically sharing security information among the users and/or client entities constituting a group.

[0040] In various embodiments, the monitored devices 102, intermediate device 212, and client network 210 may be associated with an entity, such as a business, government, or other organization, or may be associated with a single user or groups of users. That entity or those user(s) may subscribe for security services with the remote security service 204, which may monitor activities on monitored devices 102 of the entity/user(s). In some embodiments, that monitoring may involve installing agent modules 108 and situational model 110 on the monitored devices 102 to monitor execution activities 114 of the monitored devices 102 and to provide security-relevant information 118 to the remote security service 204.

[0041] The intermediate device 212 may be any sort of device configured to maintain a situational model 214 representing states of monitored devices 102 of the client. Such a situational model 214 may be like the situational model 206, but specific to a client. In some embodiments, the security modules 208 may build and query the situational model 214 in similar manner to its building and querying of the situational model 206.

Example System

[0042] FIG. 3 illustrates a component level view of a computing device capable of acting as a security service device, a monitored device, or an intermediate device. As illustrated, computing device 300 comprises a system
5 memory 302 storing modules and data 304. Also, computing device 300 includes processor(s) 306, a removable storage 308 and non-removable storage 310, input device(s) 312, output device(s) 314 and communication connections 316 for communicating with other computing devices 318.

[0043] In various embodiments, system memory 302 is volatile (such as
10 RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. The modules or data 304 stored in the system memory 302 may comprise methods, threads, processes, applications or any other sort of executable instructions, such as the instructions utilized to perform the operations of the monitored device(s) 102, remote security service 204, or
15 intermediate device 212. The modules and data 304 may also include files and databases.

[0044] In some embodiments, the processor(s) 306 is a central processing unit (CPU), a graphics processing unit (GPU), or both CPU and GPU, or other processing unit or component known in the art.

20 [0045] Computing device 300 also includes additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in FIG. 3 by removable storage 308 and non-removable storage 310. Tangible computer-

readable media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. System memory 302, removable storage 308 and non-removable storage 310 are all examples of computer-readable storage media. Computer-readable storage media include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computing device 300. Any such tangible computer-readable media may be part of the computing device 300.

[0046] Computing device 300 also has input device(s) 312, such as a keyboard, a mouse, a touch-sensitive display, voice input device, etc., and output device(s) 314 such as a display, speakers, a printer, etc. These devices are well known in the art and need not be discussed at length here.

[0047] Computing device 300 also contains communication connections 316 that allow the computing device 300 to communicate with other computing devices 318, such as others of the monitored device(s) 102, device(s) the remote security service 204, or the intermediate device 212.

Example Processes

[0048] FIGs. 4 and 5 illustrate example processes 400 and 500. These

processes are illustrated as logical flow graphs, each operation of which represents a sequence of operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the operations represent computer-executable instructions stored on one or more computer-readable storage media that, when executed by one or more processors, perform the recited operations. Generally, computer-executable instructions include routines, programs, objects, components, data structures, and the like that perform particular functions or implement particular abstract data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described operations can be combined in any order and/or in parallel to implement the processes.

[0049] FIG. 4 illustrates an example process for validating, by a monitored device, state information based on security-relevant information stored by a situational model. In some embodiments, the monitored device includes a kernel-level security agent configured to perform at least a part of the process 400.

[0050] The process 400 includes, at 402, receiving, by a monitored device, a notification of an event associated with execution activities of the monitored device.

[0051] At 404, the monitored device queries a situational model for security-relevant information based at least in part on the notification, the situational model being a substantially real-time representation of a state of the monitored device constructed from security-relevant information.

[0052] At 406, the monitored device validates state information associated with the event based at least in part on the security-relevant information. At 406a, the validating is performed conditionally based on whether the state information is fully represented in the situational model.

5 [0053] At 408, when a result of the validating is a validation error, the monitored device performs at least one security response action. At 408a, the security response action is one of reporting the validation error to a remote security service, monitoring further execution activities, taking a remediation action directed at malicious activity that is responsible for the validation error,
10 or alerting a user of the monitored device.

[0054] FIG. 5 illustrates an example process for building, by one or more computing devices, a situational model of security-relevant state for a group of monitored devices. The computing devices may be devices of a remote security service or both devices of a remote security service and an
15 intermediate device.

[0055] The process 500 includes, at 502, configuring, by one or more computing devices, a plurality of agents of a corresponding plurality of monitored devices to provide security-relevant information in substantially real-time as execution activities associated with the security-relevant
20 information are observed. In some embodiments, the security-relevant information is information that, when received from a plurality of agents, supports making security decisions.

[0056] At 504, the one or more computing devices receive the security-relevant information in substantially real-time and, at 506, build a situational model from the security-relevant information in substantially real-time as the security-relevant information is received, the situational model representing
5 states of the monitored devices. The states include at least current states of the monitored devices.

[0057] At 508, the one or more computing devices enable queries of the situational model by a user or by other modules of the remote security service. In response to receiving a query, the one or more computing devices may
10 perform at least one of actions 510-514. At 510, the one or more computing devices may validate state information for the monitored devices based at least in part on results of the queries. At 512, the one or more computing devices may perform at least one security response action based on at least in part on results of the queries. At 514, the one or more computing devices may update
15 agent configurations based at least in part on results of the queries and repeat the configuring based at least in part on the updated agent configurations.

CONCLUSION

[0058] Although the subject matter has been described in language specific
20 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. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claims.

CLAIMS

WHAT IS CLAIMED IS:

1. A system comprising:

5 one or more processors;

a memory device coupled to the one or more processors;

a situational model stored in the memory device and implemented at a kernel-level of the system, the situational model configured to include security-relevant information that represents a current state of the system and a past

10 state of the system; and

one or more modules implemented at the kernel-level and operated by the one or more processors to build the situational model with the security-relevant information in substantially real-time as execution activities associated with the security-relevant information are observed.

15

2. The system of claim 1, wherein the situational model includes a database to store the security-relevant information and corresponding metadata defining a period of time in which all data acquired about a collection of security-relevant information that includes the security-relevant information

20 has consistent properties of liveness and trustworthiness.

3. The system of claim 1, wherein the situational model includes one or more receivers to receive the security-relevant information from the one or

more modules and to store the received security-relevant information in a database of the situational model.

4. The system of claim 3, wherein the one or more receivers are
5 configured to validate state information associated with the execution activities of the system.

5. The system of claim 3, wherein the one or more receivers are
configured to generate events based on the received security-relevant information
10 and on security-relevant information stored in the database.

6. The system of claim 1, wherein the situational model includes a query interface that enables the one or more modules to access security-relevant information stored in a database of the situational model, the query interface
15 further enabling the one or more modules to specify a level of desired trustworthiness associated with metadata that corresponds to security-relevant information stored in the database.

7. The system of claim 1, wherein the situational model includes a
20 component to prune security-relevant information from the situational model based on an age and a type of security-relevant information or on a prioritization of different categories of security-relevant information.

8. The system of claim 7, wherein the component first prunes the security-relevant information based on age and type and then, if a size of the pruned situational model exceeds a limit, prunes the security-relevant information based on the prioritization of different categories.

5

9. The system of claim 1, wherein the situational model is dynamically configurable by a remote security service.

10. The system of claim 1, wherein the situational model is further
10 configured to cache at least one of information received from a remote security service, information that is computationally expensive to generate or fetch, or information that is high-latency to fetch.

11. The system of claim 1, wherein the one or more modules are
15 modules of a kernel-level security agent.

12. The system of claim 1, wherein the one or more modules are operated by the one or more processors to determine the security-relevant information based on a configuration received from a remote security service.

20

13. The system of claim 1, wherein the one or more modules are operated by the one or more processors to utilize the situational model to validate state information associated with the execution activities of the system.

14. The system of claim 1, wherein the one or more modules are operated by the one or more processors to provide security-relevant information to the remote security service.

5

15. The system of claim 1, wherein the one or more module or a remote security service utilize the security-relevant information in determining a chain of execution.

10

16. A method comprising:

receiving, by a computing device, a notification of an event associated with execution activities of the computing device;

querying, by the computing device, a situational model for security-relevant information based at least in part on the notification, the situational model being a substantially real-time representation of a state of the computing device constructed from security-relevant information; and

15

validating, by the computing device, state information associated with the event based at least in part on the security-relevant information.

20

17. The method of claim 16, wherein the validating comprises conditionally performing the validating based on whether the state information is fully maintained in the situational model.

18. The method of claim 16, wherein the receiving, the querying, and the validating are performed by a kernel-level security agent of the computing device.

5 19. The method of claim 16, further comprising, when a result of the validating is a validation error, performing at least one security response action.

20. The method of claim 19, wherein the security response action is one of reporting the validation error to a remote security service, monitoring further
10 execution activities, taking a remediation action directed at malicious activity that is responsible for the validation error, or alerting a user of the computing device.

21. One or more tangible computer-readable media having stored
15 thereon computer-executable instructions configured to program one or more computing devices to perform operations comprising:

configuring a plurality of agents of a corresponding plurality of monitored devices to provide security-relevant information in substantially real-time as execution activities associated with the security-relevant
20 information are observed;

receiving the security-relevant information in substantially real-time;
and

building a situational model from the security-relevant information in substantially real-time as the security-relevant information is received, the situational model representing states of the monitored devices.

5 22. The one or more tangible computer-readable media of claim 21, wherein the one or more computing devices implement a security service.

 23. The one or more tangible computer-readable media of claim 21, wherein the one or more devices include one or more devices implementing a
10 security service and an intermediate device, the one or more devices implementing a security service performing the configuring and the intermediate device performing the receiving and the building.

 24. The one or more tangible computer-readable media of claim 21,
15 wherein the security-relevant information is information that, when received from a plurality of agents, supports making security decisions.

 25. The one or more tangible computer-readable media of claim 21, wherein the represented states include at least current states of the monitored
20 devices.

26. The one or more tangible computer-readable media of claim 21,
wherein the operations further comprise receiving queries of the situational
model from a user or from a security service.

5 27. The one or more tangible computer-readable media of claim 26,
wherein the operations further comprise validating state information for the
monitored devices based at least in part on results of the queries.

 28. The one or more tangible computer-readable media of claim 26,
10 wherein the operations further comprise performing at least one security
response action based on at least on part on results of the queries.

 29. The one or more tangible computer-readable media of claim 26,
wherein the operations further comprise updating agent configurations based at
15 least in part on results of the queries and repeating the configuring based at
least in part on the updated agent configurations.