

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:
 - 5 receiving a first notification of a first event associated with a system component;
 - specifying a state for the system component in an event model;
 - receiving a second notification of a second event associated with a plurality of system components which include the system component; and
 - 10 based at least in part on the specified state for the system component, determining a state for the plurality of system components.
2. The method of claim 1, wherein the system components include at least one of a process, a file, or a thread.
- 15 3. The method of claim 1, wherein the first event is associated with another plurality of system components, the other plurality of system components being larger than the plurality of system components.
- 20 4. The method of claim 1, wherein the event model is a state machine and is associated with multiple state representations.

5. The method of claim 4, wherein at least one of the state representations is multi-dimensional table, each dimension being associated with a type of a system component and at least one entry in the multi-dimensional table specifying a state for a combination of system components.

5

6. The method of claim 4, wherein at least one of the first event and the second event is associated with a transition from first state to a second state.

7. The method of claim 6, further comprising determining that the transition leads from the first event to the second event by referencing a transition table associated with the event model.

8. The method of claim 1, wherein the event model is associated with a sparse table or sparse array that only specifies states for system components or combinations of system components that are associated with events.

9. The method of claim 8, wherein a state representation of the event model is associated with a wildcard table which identifies indices of instances of the event model that are to be updated responsive to an occurrence of an event.

20

10. The method of claim 9, further comprising updating the wildcard table responsive to specifying a state in a sparse table or sparse array for a system component or a combination of system components that was not previously associated with an event.

5

11. The method of claim 8, further comprising updating states in the sparse table responsive to receiving the notification of the first event, the updating including updating states in a row or a column of the sparse table.

10

12. The method of claim 1, wherein the determining includes referencing state representations of the event model and a transition table associated with the event model.

13. One or more non-transitory computer-readable media having stored thereon computer-executable instructions which, when executed by a computing device, cause the computing device to perform operations comprising:

receiving a first notification of a first event associated with a system component;

specifying a state for the system component in an event model;

receiving a second notification of a second event associated with a plurality of system components which include the system component; and

20

based at least in part on the specified state for the system component,
determining a state for the plurality of system components.

14. The one or more non-transitory computer-readable media of claim
5 13, wherein the operations further comprise generating an event responsive to
the first event or the second event and dispatching the generated event to an
event bus.

15. The one or more non-transitory computer-readable media of claim
10 13, wherein the operations further comprise identifying a transition associated
with the first event or the second event, retrieving an action list associated with
the transition in a transition table of the event model, and performing at least
one action specified in the action list.

15 16. A system comprising:

one or more processors;

a kernel-level security agent configured to be operated by the one or
more processors, the kernel-level security agent including:

an event bus configured to receive and dispatch events associated
20 with system components;

an event model configured to maintain states for system
components and combinations of system components; and

an event model manager configured to receive notifications of events from the event bus and to determine a state for a combination of components based at least in part on an event and on a state for one component of the combination of system components that is maintained by the event model.

5

17. The system of claim 16, wherein the event bus provides a parameter with an event when dispatching that event to the event model manager.

10

18. The system of claim 17, wherein the parameter specifies A) an event model, B) which instances of the event model are to be updated based on the event, and C) a transition from one state in the event model to another state in the event model.

15

19. The system of claim 17, wherein the event model manager is configured to determine the state based on the parameter.

20

20. The system of claim 17, wherein the event bus determines the parameter based on a configuration of the kernel-level security agent.

21. The system of claim 20, wherein the event bus further determines the parameter based on the event passing one or more filters and dispatches the event to the event model manager multiple times with multiple different parameters.

5

22. The system of claim 20, wherein at least a component of the event model is updated in response to an updated configuration of the kernel-level security agent without any reboot of the system or recompile of the kernel-level security agent.

10

23. The system of claim 16, wherein the kernel-level security agent is configured to perform at least one of monitoring events on a computing device, maintaining state about some or all of the events on a computing device, notifying a security service of one or more events on a computing device, or taking action responsive to a security exploit associated with one or more events on a computing device.

15

24. The system of claim 16, wherein the states in the event model are associated with name_value pairs which describe information associated with events.

20

25. The system of claim 16, further comprising an accumulator associated with the event model, the accumulator maintaining a count of occurrences of a specific event, wherein the event model manager is further configured to determine the state based at least in part on the count maintained
5 by the accumulator.

26. The system of claim 16, further comprising an accumulator associated with the event model, the accumulator maintaining a value that is updatable responsive to occurrence of one or more events, wherein the event
10 model manager is further configured to determine the state based at least in part on the value maintained by the accumulator.

CLAIMS

1. A computer-implemented method comprising:
receiving a first notification of a first event associated with a system component;
specifying a state for the system component in an event model;
5 receiving a second notification of a second event associated with a plurality of system components which include the system component; and
based at least in part on the specified state for the system component, determining a state for the plurality of system components.
- 10 2. The method of claim 1, wherein the system components include at least one of a process, a file, or a thread.
3. The method of claim 1, wherein the first event is associated with another plurality of system components, the other plurality of system components being larger than the plurality of
15 system components.
4. The method of claim 1, wherein the event model is a state machine and is associated with multiple state representations.
- 20 5. The method of claim 4, wherein at least one of the state representations is multi-dimensional table, each dimension being associated with a system component type of and at least one entry in the multi-dimensional table specifying a state for a combination of system components.

6. The method of claim 4, wherein at least one of the first event and the second event is associated with a transition from first state to a second state.

5 7. The method of claim 6, further comprising determining that the transition leads from the first event to the second event by referencing a transition table associated with the event model.

8. The method of claim 1, wherein the event model is associated with a sparse table or
10 sparse array that only specifies states for system components or combinations of system components that are associated with events.

9. The method of claim 8, wherein a state representation of the event model is associated with a wildcard table which identifies indices of instances of the event model that are to be
15 updated responsive to an occurrence of an event.

10. The method of claim 9, further comprising updating the wildcard table responsive to specifying a state in a sparse table or sparse array for a system component or a combination of system components that was not previously associated with an event.

20

11. The method of claim 8, further comprising updating states in the sparse table responsive to receiving the notification of the first event, the updating including updating states in a row or a column of the sparse table.

12. The method of claim 1, wherein the determining includes referencing state representations of the event model and a transition table associated with the event model.

5 13. One or more non-transitory computer-readable media having stored thereon computer-executable instructions which, when executed by a computing device, cause the computing device to perform operations comprising:

receiving a first notification of a first event associated with a system component;

specifying a state for the system component in an event model;

10 receiving a second notification of a second event associated with a plurality of system components which include the system component; and

based at least in part on the specified state for the system component, determining a state for the plurality of system components.

15 14. The one or more non-transitory computer-readable media of claim 13, wherein the operations further comprise generating an event responsive to the first event or the second event and dispatching the generated event to an event bus.

20 15. The one or more non-transitory computer-readable media of claim 13, wherein the operations further comprise identifying a transition associated with the first event or the second event, retrieving an action list associated with the transition in a transition table of the event model, and performing at least one action specified in the action list.

16. A system comprising:

one or more processors;

a kernel-level security agent configured to be operated by the one or more processors, the kernel-level security agent including:

5 an event bus configured to receive and dispatch events associated with system components;

 an event model configured to maintain states for system components and combinations of system components; and

 an event model manager configured to receive notifications of events from the
10 event bus and to determine a state for a combination of components based at least in part on an event and on a state for one component of the combination of system components that is maintained by the event model.

17. The system of claim 16, wherein the event bus provides a parameter with an event
15 when dispatching that event to the event model manager.

18. The system of claim 17, wherein the parameter specifies A) the event model, B) which instances of the event model are to be updated based on the event, and C) a transition from one state in the event model to another state in the event model.
20

19. The system of claim 17, wherein the event model manager is configured to determine the state based on the parameter.

20. The system of claim 17, wherein the event bus determines the parameter based on a configuration of the kernel-level security agent.

21. The system of claim 20, wherein the event bus further determines the parameter based
5 on the event passing one or more filters and dispatches the event to the event model manager multiple times with multiple different parameters.

22. The system of claim 20, wherein at least a component of the event model is updated
in response to an updated configuration of the kernel-level security agent without any reboot of
10 the system or recompile of the kernel-level security agent.

23. The system of claim 16, wherein the kernel-level security agent is configured to
perform at least one of monitoring events on a computing device, maintaining state about some
or all of the events on the computing device, notifying a security service of one or more events
15 on the computing device, or taking action responsive to a security exploit associated with one or
more events on the computing device.

24. The system of claim 16, wherein the states in the event model are associated with
name_value pairs which describe information associated with events.
20

25. The system of claim 16, further comprising an accumulator associated with the event
model, the accumulator maintaining a count of occurrences of a specific event, wherein the event

model manager is further configured to determine the state based at least in part on the count maintained by the accumulator.

26. The system of claim 16, further comprising an accumulator associated with the event
5 model, the accumulator maintaining a value that is updatable responsive to occurrence of one or more events, wherein the event model manager is further configured to determine the state based at least in part on the value maintained by the accumulator.