(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 792 912**

(51) Int. CI.:

G06F 21/55 (2013.01)
G06F 21/53 (2013.01)
G06F 21/54 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **14.07.2016 PCT/EP2016/066745**

(87) Fecha y número de publicación internacional: **19.01.2017 WO17009415**

(96) Fecha de presentación y número de la solicitud europea: **14.07.2016 E 16741900 (1)**

(97) Fecha y número de publicación de la concesión europea: **18.03.2020 EP 3323074**

(54) Título: **Sistemas y métodos de seguridad informática que utilizan excepciones de introspección asíncronas**

(30) Prioridad:

14.07.2015 US 201562192384 P
13.07.2016 US 201615209317

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
12.11.2020

(73) Titular/es:

BITDEFENDER IPR MANAGEMENT LTD. (100.0%)
Kreontos 12
1076 Nicosia, CY

(72) Inventor/es:

LUKACS, SANDOR;
SIRB, CRISTIAN-BOGDAN y
LUTAS, ANDREI-VLAD

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Sistemas y métodos de seguridad informática que utilizan excepciones de introspección asíncronas

Solicitudes relacionadas

Antecedentes

La invención se refiere a sistemas y métodos de seguridad informática, y, en particular, a sistemas y métodos para proteger entornos de virtualización de *hardware* contra amenazas de seguridad informática.

El *software* malicioso, también conocido como *malware*, afecta a una gran cantidad de sistemas informáticos en todo el mundo. En sus muchas formas, tales como virus informáticos, gusanos informáticos, *rootkits*, *software* espía y *software* publicitario no deseado, el *malware* presenta un grave riesgo para millones de usuarios de ordenadores, haciendo que los mismos sean vulnerables a la pérdida de datos e información confidencial, a robos de identidad y a la pérdida de productividad, entre otros.

El *software* de seguridad informática puede usarse para proteger sistemas informáticos contra *software* malicioso. Los métodos comúnmente utilizados para detectar y combatir el *malware* incluyen métodos de comparación de firmas y basados en el comportamiento. Los métodos basados en firmas intentan encontrar una coincidencia de una sección de código de una entidad de *software* objetivo con una colección de fragmentos de código extraídos de *software* del cual se sabe que es malicioso. Los métodos basados en el comportamiento generalmente comprenden detectar la manifestación de un evento provocado por o que se produce durante la ejecución de una entidad de *software* objetivo, y analizar el evento respectivo para determinar si el mismo indica una amenaza potencial a la seguridad.

La detección convencional de eventos se basa típicamente en una clase de métodos conocidos en la técnica como *hooking*. Dichos métodos a menudo son vulnerables y pueden verse frustrados por *software* malicioso. Además, los métodos convencionales basados en el comportamiento generalmente suspenden la ejecución de la entidad que provocó un evento detectado, mientras el evento respectivo se analiza para detectar indicadores de malicia. Dichas suspensiones pueden afectar negativamente a la experiencia del usuario, especialmente en configuraciones de virtualización de *hardware* en las que se ejecuta *software* de seguridad fuera de una máquina virtual protegida.

Existe un interés continuo en mejorar la eficiencia de los sistemas y métodos de seguridad informática, y en particular en desarrollar sistemas y métodos que aborden las deficiencias anteriores relacionadas con la detección y el análisis de eventos. El documento de patente US2012/255012 representa técnica anterior relevante.

Sumario

Según un aspecto, un sistema anfitrión comprende un procesador de *hardware* y una memoria, estando configurado el procesador de *hardware* para ejecutar una entidad objetivo, un analizador de excepciones síncronas y un analizador de excepciones asíncronas. El procesador de *hardware* está configurado, además, en respuesta a la detección de una manifestación de un evento provocado por una ejecución de la entidad objetivo, para suspender la ejecución de la entidad objetivo, y en respuesta a la suspensión de la ejecución de la entidad objetivo, para cambiar a ejecutar el analizador de excepciones síncronas. El analizador de excepciones síncronas está configurado para determinar si la entidad objetivo es sospechosa de ser maliciosa según el evento. El analizador de excepciones síncronas está configurado además, en respuesta, cuando la entidad objetivo es sospechosa de ser maliciosa, para recuperar selectivamente una firma de excepción de regla de una pluralidad de firmas de excepción de regla, recuperándose la firma de excepción de regla de acuerdo con el evento, en donde la firma de excepción de regla comprende una codificación de una primera condición y una codificación de una segunda condición. El analizador de excepciones síncronas está configurado, además, en respuesta a la recuperación de la firma de excepción, para determinar si la primera condición se cumple de acuerdo con el evento y de acuerdo con la entidad objetivo. En respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, el analizador de excepciones síncronas está configurado además para hacer que el procesador de *hardware* reanude la ejecución de la entidad objetivo. El analizador de excepciones síncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la primera condición, cuando la primera condición no se cumple, para determinar que la entidad objetivo es maliciosa. El analizador de excepciones asíncronas está configurado, en respuesta a la reanudación de la ejecución de la entidad objetivo por parte del procesador de *hardware*, para determinar si se cumple la segunda condición de acuerdo con el evento y de acuerdo con la entidad objetivo. El analizador de excepciones asíncronas está configurado, además, en respuesta a la determinación de si se cumple la segunda condición, cuando se cumple la segunda condición, para determinar que la entidad objetivo no es maliciosa. El analizador de excepciones asíncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la segunda condición, cuando la segunda condición no se cumple, para determinar que la entidad objetivo es maliciosa.

Según otro aspecto, un soporte no transitorio legible por ordenador almacena instrucciones de procesador que, cuando son ejecutadas por un procesador de *hardware* de un sistema anfitrión, hacen que el sistema anfitrión forme un analizador de excepciones síncronas y un analizador de excepciones asíncronas. El procesador de *hardware* está configurado, además, en respuesta a la detección de una manifestación de un evento provocado por una ejecución de la entidad objetivo, para suspender la ejecución de la entidad objetivo, y en respuesta a la suspensión de la

ejecución de la entidad objetivo, para cambiar a ejecutar el analizador de excepciones síncronas. El analizador de excepciones síncronas está configurado para determinar si la entidad objetivo es sospechosa de ser maliciosa según el evento. El analizador de excepciones síncronas está configurado, además, en respuesta, cuando la entidad objetivo es sospechosa de ser maliciosa, para recuperar selectivamente una firma de excepción de regla de una pluralidad de firmas de excepción de regla, recuperándose la firma de excepción de regla de acuerdo con el evento, en donde la firma de excepción de regla comprende una codificación de una primera condición y una codificación de una segunda condición. El analizador de excepciones síncronas está configurado, además, en respuesta a la recuperación de la firma de excepción, para determinar si se cumple la primera condición de acuerdo con el evento y de acuerdo con la entidad objetivo. En respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, el analizador de excepciones síncronas está configurado además para hacer que el procesador de *hardware* reanude la ejecución de la entidad objetivo. El analizador de excepciones síncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la primera condición, cuando la primera condición no se cumple, para determinar que la entidad objetivo es maliciosa. El analizador de excepciones asíncronas está configurado, en respuesta a que el procesador de *hardware* reanude la ejecución de la entidad objetivo, para determinar si se cumple la segunda condición de acuerdo con el evento y de acuerdo con la entidad objetivo. El analizador de excepciones asíncronas está configurado además, en respuesta a la determinación de si se cumple la segunda condición, cuando se cumple la segunda condición, para determinar que la entidad objetivo no es maliciosa. El analizador de excepciones asíncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la segunda condición, cuando la segunda condición no se cumple, para determinar que la entidad objetivo es maliciosa.

Según otro aspecto, un método protege un sistema anfitrión contra amenazas de seguridad informática, en el que el sistema anfitrión comprende un procesador de *hardware* y una memoria. El método comprende utilizar el procesador de *hardware* para detectar una manifestación de un evento provocado por una ejecución de una entidad objetivo. El método comprende, además, en respuesta a la detección de la manifestación del evento, utilizar el procesador de *hardware* para suspender la ejecución de la entidad objetivo y cambiar a ejecutar un analizador de excepciones síncronas. El analizador de excepciones síncronas está configurado para determinar si la entidad objetivo es sospechosa de malicia según el evento. En respuesta, cuando la entidad objetivo es sospechosa de malicia, el analizador de excepciones síncronas está configurado para recuperar selectivamente una firma de excepción de regla de una pluralidad de firmas de excepción de regla, recuperándose la firma de excepción de regla de acuerdo con el evento, en donde la firma de excepción de regla comprende una codificación de una primera condición y una codificación de una segunda condición. El analizador de excepciones síncronas está configurado adicionalmente, en respuesta a la recuperación de la firma de excepción de regla, para determinar si se cumple la primera condición según el evento y según la entidad objetivo. El analizador de excepciones síncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, para hacer que el procesador de *hardware* reanude la ejecución de la entidad objetivo, y cuando la primera condición no se cumple, para determinar que la entidad objetivo es maliciosa. El método comprende además, en respuesta a que el procesador de *hardware* reanude la ejecución de la entidad objetivo, utilizar el procesador de *hardware* para determinar si se cumple la segunda condición según el evento y según la entidad objetivo. El método comprende, además, en respuesta a la determinación de si se cumple la segunda condición, cuando se cumple la segunda condición, determinar que la entidad objetivo no es maliciosa, y cuando la segunda condición no se cumple, determinar que la entidad objetivo es maliciosa.

Breve descripción de los dibujos

Los aspectos y ventajas anteriores de la presente invención se entenderán mejor al leer la siguiente descripción detallada y en referencia a los dibujos donde:

La Fig. **1** ilustra una configuración de *hardware* ejemplificativa de un sistema anfitrión protegido contra amenazas de seguridad informática de acuerdo con algunas realizaciones de la presente invención.

La Fig. **2-A** muestra una aplicación de seguridad informática (CSA) ejemplificativa que protege el sistema anfitrión de acuerdo con algunas realizaciones de la presente invención, en una configuración que no implica virtualización de *hardware*.

La Fig. **2-B** muestra una configuración alternativa de acuerdo con algunas realizaciones de la presente invención, en la que un conjunto ejemplificativo de máquinas virtuales protegidas es expuesto por un hipervisor que se ejecuta en el sistema anfitrión, y en el que la CSA se ejecuta fuera de la(s) máquina(s) virtual(es) protegida(s).

La Fig. **3** ilustra componentes ejemplificativos de una aplicación de seguridad informática según algunas realizaciones de la presente invención.

La Fig. **4-A** muestra una configuración ejemplificativa, en la que el administrador de notificaciones se ejecuta dentro de la máquina virtual protegida, y en la que los analizadores de excepciones síncronas y asíncronas se ejecutan fuera de la máquina virtual protegida.

La Fig. **4-B** muestra una configuración alternativa de acuerdo con algunas realizaciones de la presente invención, en la que el administrador de notificaciones se ejecuta fuera de la máquina virtual protegida, y en la que los analizadores

de excepciones síncronas y asíncronas se ejecutan dentro de la máquina virtual protegida.

La Fig. **4-C** muestra aún otra configuración ejemplificativa según algunas realizaciones de la presente invención, en la que el analizador de excepciones asíncronas se ejecuta dentro de una máquina virtual de seguridad distinta de la máquina virtual protegida.

La Fig. **5** muestra una interacción ejemplificativa de los componentes de la aplicación de seguridad informática según algunas realizaciones de la presente invención.

La Fig. **6** ilustra un formato ejemplificativo de una excepción de regla según algunas realizaciones de la presente invención.

La Fig. **7** muestra un formato ejemplificativo de una solicitud de análisis de excepción (EAR) de acuerdo con algunas realizaciones de la presente invención.

La Fig. **8** muestra una secuencia ejemplificativa de pasos llevados a cabo por el administrador de notificaciones de acuerdo con algunas realizaciones de la presente invención.

La Fig. **9** ilustra una secuencia ejemplificativa de pasos realizados por el analizador de excepciones síncronas de acuerdo con algunas realizaciones de la presente invención.

La Fig. **10** muestra una secuencia ejemplificativa de pasos llevados a cabo por el analizador de excepciones asíncronas de acuerdo con algunas realizaciones de la presente invención.

La Fig. **11** ilustra una secuencia ejemplificativa de pasos realizados por el supervisor de terminación de acuerdo con algunas realizaciones de la presente invención.

Descripción detallada de realizaciones preferidas

En la siguiente descripción, se entiende que todas las conexiones mencionadas entre estructuras pueden ser conexiones operativas directas o conexiones operativas indirectas a través de estructuras intermedias. Un conjunto de elementos incluye uno o más elementos. Se entiende que cualquier mención de un elemento se refiere al menos a un elemento. Una pluralidad de elementos incluye al menos dos elementos. A menos que se requiera lo contrario, los pasos descritos del método no tienen que realizarse necesariamente en un orden ilustrado particular. Un primer elemento (por ejemplo, datos) derivado de un segundo elemento abarca un primer elemento igual al segundo elemento, así como un primer elemento generado al procesar el segundo elemento y opcionalmente otros datos. Tomar una determinación o decisión de acuerdo con un parámetro incluye tomar la determinación o decisión de acuerdo con el parámetro y opcionalmente de acuerdo con otros datos. A menos que se especifique lo contrario, un indicador de cierta cantidad/datos puede ser la propia cantidad/datos, o un indicador diferente de la propia cantidad/datos. La seguridad informática abarca la protección de usuarios y equipos contra un acceso no deseado o no autorizado a datos y/o *hardware*, una modificación no deseada o no autorizada de datos y/o *hardware*, y una destrucción de datos y/o *hardware*. Un programa de ordenador es una secuencia de instrucciones de procesador que realiza una tarea. Los programas de ordenador descritos en algunas realizaciones de la presente invención pueden ser entidades o subentidades de *software* independientes (por ejemplo, subrutinas, bibliotecas) de otros programas de ordenador. A menos que se especifique lo contrario, un proceso es una instancia de un programa de ordenador, tal como una aplicación o una parte de un sistema operativo, y se caracteriza por tener al menos un hilo de ejecución y un espacio de memoria virtual asignado al mismo, en el que un contenido del espacio de memoria virtual respectivo incluye código ejecutable. A menos que se especifique lo contrario, un proceso invitado es un proceso que se ejecuta dentro de una máquina virtual. Se dice que un proceso se ejecuta dentro de una máquina virtual cuando se ejecuta en un procesador virtual de la máquina virtual respectiva. A menos que se especifique lo contrario, una página representa la unidad más pequeña de memoria virtual que se puede asignar individualmente a una memoria física de un sistema anfitrión. Los soportes legibles por ordenador incluyen soportes no transitorios, tales como soportes de almacenamiento magnéticos, ópticos y de semiconductores (por ejemplo, unidades de disco duro, discos ópticos, memoria *flash*, DRAM), así como enlaces de comunicación, tales como cables conductores y enlaces de fibra óptica. Según algunas realizaciones, la presente invención proporciona, entre otros, sistemas informáticos que comprenden *hardware* (por ejemplo, uno o más microprocesadores) programados para realizar los métodos descritos en la presente, así como soportes legibles por ordenador que codifican instrucciones para realizar los métodos descritos en la presente.

La siguiente descripción ilustra realizaciones de la invención a modo de ejemplo y no necesariamente a modo de limitación.

La Fig. **1** muestra una configuración de *hardware* ejemplificativa de un sistema anfitrión **10** protegido contra amenazas de seguridad informática según algunas realizaciones de la presente invención. El sistema anfitrión **10** puede representar cualquier dispositivo electrónico que tenga un procesador y una memoria. Sistemas anfitriones **10** ejemplificativos incluyen ordenadores personales, servidores, ordenadores portátiles, ordenadores de tipo tableta, dispositivos de telecomunicaciones móviles (por ejemplo, teléfonos inteligentes), reproductores multimedia, televisores, consolas de juegos, electrodomésticos (por ejemplo, neveras, termostatos, sistemas inteligentes de

calefacción y/o iluminación) y dispositivos ponibles (por ejemplo, relojes inteligentes, equipos deportivos y de *fitness*), entre otros.

La Fig. **1** ilustra un sistema informático; la configuración de *hardware* de otros sistemas anfitriones, tales como teléfonos inteligentes y relojes inteligentes, puede diferir de la configuración ilustrada. El sistema anfitrión **10** comprende un conjunto de dispositivos físicos, incluidos un procesador **12** y una unidad **14** de memoria**.** En algunas realizaciones, el procesador **12** comprende un dispositivo físico (por ejemplo, un microprocesador, un circuito integrado multinúcleo formado en un sustrato semiconductor, etc.) configurado para ejecutar operaciones computacionales y/o lógicas con un conjunto de señales y/o datos. En algunas realizaciones, tales operaciones se entregan al procesador **12** en forma de una secuencia de instrucciones de procesador (por ejemplo, código máquina u otro tipo de codificación). La unidad **14** de memoria puede comprender soportes volátiles legibles por ordenador (por ejemplo, DRAM, SRAM) que almacenan instrucciones y/o datos a los que accede el procesador **12** ó que son generados por este último.

Dependiendo del tipo y del rendimiento del dispositivo, el sistema anfitrión **10** puede comprender además un conjunto de dispositivos **16** de entrada, tales como un teclado, un ratón, una pantalla táctil, etc., que permiten a un usuario introducir datos y/o instrucciones en el sistema anfitrión **10.** Un conjunto de dispositivos **18** de salida, tales como un monitor o pantalla de cristal líquido, puede transmitir información al usuario, por ejemplo, a través de una interfaz gráfica de usuario. Los dispositivos **20** de almacenamiento incluyen soportes legibles por ordenador que permiten el almacenamiento no volátil, la lectura y la escritura de instrucciones y/o datos del procesador. Dispositivos **20** de almacenamiento ejemplificativos incluyen discos magnéticos y ópticos y dispositivos de memoria *flash*, así como soportes extraíbles, tales como discos y unidades de CD y/o DVD. El conjunto de adaptadores **22** de red permite que el sistema anfitrión **10** se conecte a una red informática y/o a otros dispositivos/sistemas informáticos. El concentrador controlador **24** representa genéricamente la pluralidad de buses del sistema, periféricos y/o de conjuntos de chips, y/o la totalidad del resto de circuitos que permiten la comunicación entre el procesador **12** y los dispositivos **14**, **16**, **18**, **20** y **22.** Por ejemplo, el concentrador controlador **24** puede incluir una unidad de administración de memoria (MMU), un controlador de entrada/salida (E/S) y un controlador de interrupciones, entre otros. En otro ejemplo, el concentrador controlador **24** puede comprender un procesador **12** de conexión de puente norte (*northbridge*) con la memoria **14** y/o un procesador **12** de conexión de puente sur (*southbridge*) con los dispositivos **16**, **18**, **20** y **22.** En algunas realizaciones, el concentrador controlador **24** puede integrarse, parcial o totalmente, con el procesador **12**, por ejemplo, la MMU puede compartir un sustrato semiconductor común con el procesador **12.**

Una aplicación de seguridad informática (CSA) **40** protege el sistema anfitrión **10** contra amenazas de seguridad informática, tales como *malware*, *software* espía, *software* publicitario no deseado, etc. En algunas realizaciones, la CSA **40** está configurada para monitorizar el comportamiento de una pluralidad de entidades ejecutables (por ejemplo, procesos, hilos, aplicaciones, componentes del sistema operativo) y para determinar si alguna de esas entidades monitorizadas es maliciosa de acuerdo con su comportamiento. Las entidades monitorizadas pueden incluir componentes de un sistema operativo y aplicaciones de usuario, entre otros. En respuesta a la determinación de que una entidad es maliciosa, la CSA **40** puede llevar a cabo una acción de protección contra la entidad respectiva, por ejemplo para detener, poner en cuarentena o incapacitar de otra manera la entidad maliciosa respectiva.

Las Figs. **2-A-B** muestran configuraciones de *software* ejemplificativas de acuerdo con algunas realizaciones de la presente invención. En el ejemplo de la Fig. **2-A**, el sistema anfitrión **10** ejecuta un sistema operativo (OS) **34**, un conjunto de aplicaciones ejemplificativas **36a-b** y la CSA **40.** Las aplicaciones **36a-b** representan genéricamente cualquier programa de ordenador, tal como aplicaciones de procesado de texto, de procesado de imágenes, de reproductor multimedia, de base de datos, de calendario, de administración de contactos personales, de navegador, de juego, de comunicación de voz y de comunicación de datos, entre otras. El sistema operativo **34** puede comprender cualquier sistema operativo ampliamente disponible tal como Microsoft Windows®, MacOS®, Linux®, iOS® o Android®, entre otros. El OS **34** proporciona una interfaz entre aplicaciones **36a-b** y el *hardware* del sistema anfitrión **10.** La posición ilustrada de la CSA **40** indica que la CSA **40** puede ejecutarse en varios niveles de privilegio del procesador. Por ejemplo, una parte de la CSA **40** puede ejecutarse en el privilegio del procesador del *kernel* del OS (por ejemplo, anillo 0, modo *kernel*), mientras que otras partes pueden ejecutarse en el privilegio del procesador de las aplicaciones **36a-b** (p. ej., anillo 3, modo de usuario).

La Fig. **2-B** muestra una realización alternativa de la presente invención, en la que el sistema anfitrión **10** utiliza tecnología de virtualización de *hardware* para operar un conjunto de máquinas virtuales invitadas. La virtualización de *hardware* a menudo se usa en aplicaciones tales como la computación en la nube y la consolidación de servidores, entre otros usos. Una máquina virtual (VM) emula una máquina/sistema informático físico real, y es capaz de ejecutar un sistema operativo y otras aplicaciones. En algunas realizaciones, un hipervisor **30** se ejecuta en el sistema anfitrión **10**, estando configurado el hipervisor **30** para crear o habilitar una pluralidad de dispositivos virtualizados, tales como un procesador virtual y una unidad de administración de memoria virtual, y para presentar dichos dispositivos virtualizados a otro *software*, en lugar de los dispositivos físicos reales del sistema anfitrión **10.** Dichas operaciones se conocen comúnmente en la técnica como exposición de una máquina virtual. El hipervisor **30** puede permitir además que varias máquinas virtuales compartan los recursos de *hardware* del sistema anfitrión **10**, para que cada VM funcione de manera independiente y no tenga conocimiento de que otras VM se ejecutan simultáneamente en el sistema anfitrión **10.** Los ejemplos de hipervisores populares incluyen el VMware vSphere ™ de VMware Inc. y el hipervisor de código abierto Xen, entre otros.

La Fig. **2-B** muestra un conjunto de VMs invitadas **32a-b** expuesto en el sistema anfitrión **10.** Cada VM **32a-b** incluye un procesador virtualizado y puede incluir además otros dispositivos virtualizados, tales como dispositivos de entrada, de salida, de almacenamiento y de red virtualizados, así como un controlador virtualizado, entre otros. Cada procesador virtualizado comprende una emulación de al menos parte de la funcionalidad del procesador **12** de *hardware* y está configurado para recibir instrucciones de procesador para su ejecución. Se dice que el *software* que usa el procesador virtual respectivo para la ejecución se ejecuta dentro de la máquina virtual respectiva. Por ejemplo, en el ejemplo de la Fig. **2-B**, se dice que el OS invitado **34a** y la aplicación **36c** se ejecutan dentro de la VM invitada **32a.** Por contraposición, se dice que el hipervisor **30** se ejecuta fuera, o por debajo, de las VMs invitadas **32a-b.** Cada procesador virtualizado puede interactuar con dichos dispositivos virtualizados como lo haría con los dispositivos físicos correspondientes. Por ejemplo, el *software* que se ejecuta dentro de la VM **32a** puede enviar y/o recibir tráfico de red a través de un(os) adaptador(es) de red virtualizado(s) de la VM **32a.** En algunas realizaciones, el hipervisor **30** puede exponer solo un subconjunto de dispositivos virtualizados a cada VM invitada, y puede conceder a una VM invitada seleccionada el uso directo y exclusivo de algunos dispositivos de *hardware* del sistema anfitrión **10.** En uno de esos ejemplos, la VM **32a** puede tener uso exclusivo de dispositivos **16** de entrada y dispositivos **18** de salida, pero carecer de un adaptador de red virtualizado. Al mismo tiempo, la VM **32b** puede tener uso directo y exclusivo del(de los) adaptador(es) **22** de red**.** Dichas configuraciones pueden implementarse, por ejemplo, utilizando la tecnología VT-d® de Intel®.

Cada VM **32a-b** ejecuta un sistema operativo (OS) invitado **34a-b**, respectivamente. Cada OS **34a-b** proporciona una interfaz entre aplicaciones **36c-d** que se ejecutan dentro de la VM respectiva y los dispositivos de *hardware* virtualizados de la VM respectiva. En la realización ejemplificativa de las Figs. **2-B**, la CSA **40** se ejecuta fuera de las VMs invitadas y está configurada para proteger las VMs invitadas respectivas contra las amenazas de seguridad informática. Una sola aplicación de seguridad informática puede proteger una pluralidad de máquinas virtuales. La CSA **40** puede incorporarse al hipervisor **30**, por ejemplo, como una biblioteca, o puede entregarse como un programa de ordenador distinto e independiente del hipervisor **30**, pero ejecutándose en el nivel de privilegio de procesador del hipervisor **30** (por ejemplo, modo raíz, anillo -1). La CSA **40** puede ser un proceso que tenga un hilo de ejecución planificado por separado, o puede funcionar como una colección de objetos de código no planificados que se ejecutan cuando son activados por ciertos eventos de notificación, como se ilustra más adelante.

En la técnica se conocen varios métodos para proteger, del *malware,* sistemas anfitriones, incluidas las plataformas de virtualización de *hardware*. Una categoría particular de métodos se conoce como análisis de comportamiento. Los métodos de análisis de comportamiento típicos utilizan un mecanismo de notificación, en el que al *software* de seguridad se le notifica la manifestación de un evento dentro de una VM monitorizada, en donde el evento es activado por y/o se produce durante la ejecución de una entidad de *software*, tal como una aplicación o componente del sistema operativo. El *software* de seguridad puede analizar entonces el evento respectivo para determinar si el mismo indica una amenaza potencial a la seguridad.

La Fig. **3** muestra componentes ejemplificativos de la aplicación **40** de seguridad informática de acuerdo con algunas realizaciones de la presente invención. El motor **40** incluye un administrador **42** de notificaciones, un analizador **44** de excepciones síncronas conectado al administrador **42**, un analizador **46** de excepciones asíncronas y un supervisor **48** de terminación acoplado comunicativamente al analizador **46.**

En algunas realizaciones, el administrador **42** de notificaciones está configurado para detectar la manifestación de ciertos eventos relevantes para la seguridad informática. Los ejemplos de eventos detectados incluyen, entre otros, llamadas a ciertas funciones del OS y llamadas del sistema. Otros tipos de eventos detectados pueden incluir abrir un archivo, crear un archivo, escribir en un archivo, eliminar un archivo, copiar un archivo, crear un proceso, finalizar un proceso, planificar un hilo para su ejecución, suspender un hilo debido a un evento de sincronización (por ejemplo, exclusión mutua), crear un *heap*, asignar memoria del *heap*, ampliar el tamaño de una pila de ejecución, cambiar un permiso de acceso a la memoria, realizar una operación de intercambio de entrada (por ejemplo, de disco a memoria), realizar una operación de intercambio de salida (por ejemplo, de memoria a disco), cargar un módulo ejecutable (por ejemplo, biblioteca compartida - DLL), abrir una clave de registro, cambiar el nombre de una clave de registro, detectar la conexión de un nuevo dispositivo de *hardware*, establecer una nueva conexión de red, recibir un paquete de red, elevar los privilegios de ejecución de un hilo, cambiar el permiso de control de acceso discrecional (DAC) asociado a un archivo. En la técnica se conocen varios métodos para detectar tales eventos. Los mismos incluyen aplicar ganchos (*hooking*) a ciertas funciones del OS, modificar tablas de despacho, etc. Tales métodos configuran el procesador **12** para cambiar de ejecutar la entidad desencadenante (por ejemplo, un proceso) a ejecutar una rutina de administrador en respuesta a una manifestación del evento respectivo. El registro del administrador **42** de notificaciones como rutina de administrador permite al administrador **42** detectar varios eventos y comunicar su manifestación a la CSA **40.**

En plataformas de virtualización de *hardware*, una categoría especial de eventos detectados que puede ser relevantes para la seguridad informática incluye la detección de una violación de un permiso de acceso a la memoria. La detección de tales violaciones puede proporcionar una alternativa a la aplicación de ganchos (*hooking*) convencional. La mayoría de los sistemas informáticos modernos están configurados para funcionar con memoria virtual y para administrar las traducciones de direcciones de memoria utilizando estructuras de datos dedicadas, por ejemplo, tablas de páginas. Los sistemas configurados para admitir la virtualización de *hardware* usan típicamente una segunda capa de traducciones de direcciones, desde una memoria física de invitado vista por cada VM expuesta a la memoria física real **14** del sistema anfitrión. La segunda traducción de direcciones se logra típicamente utilizando mecanismos y

estructuras de datos, dedicados acelerados por *hardware,* y controlados por el procesador **12**, conocidos como traducción de direcciones de segundo nivel (SLAT). Las implementaciones populares de la SLAT incluyen tablas de páginas extendidas (EPT) en plataformas Intel®, y la indexación rápida de virtualización (RVI)/tablas de páginas anidadas (NPT) en plataformas AMD®. La SLAT permite típicamente configurar permisos de acceso a memoria para cada página de memoria, tales como lectura/escritura/ejecución. El procesador **12** puede configurarse para activar un evento de procesador (por ejemplo, un evento de salida de VM o una excepción de virtualización) cuando el *software* intenta acceder a la página respectiva de una manera que viola los permisos de acceso en curso. Los eventos de salida de VM, por ejemplo VMExit en plataformas Intel®, suspenden la ejecución de código dentro de la VM respectiva y hacen que el procesador **12** cambie a la ejecución de código en el nivel del hipervisor **30**. Por el contrario, las excepciones de virtualización, tales como #VE en plataformas Intel®, pueden hacer que el procesador **12** cambie a la ejecución de código dentro de la misma VM. En algunas realizaciones, la CSA **40** registra el administrador **42** de notificaciones como administrador para salidas de VM (por ejemplo, en configuraciones tales como la Fig. **4-B**) o excepciones de virtualización (p. ej., en configuraciones tales como las Figs. **4-A** y **4-C**)**.** Esto permite al administrador **42** detectar intentos de violar permisos de acceso a la memoria dentro de una máquina virtual protegida, y comunicar dichos intentos a la CSA **40.**

Los sistemas de seguridad convencionales a menudo se basan en firmas de comportamiento de *malware* para detectar entidades maliciosas. Una firma de comportamiento de *malware* comprende un conjunto de condiciones que, cuando son satisfechas por un evento (o secuencia de eventos), indican que la entidad que activa el(los) evento(s) respectivo(s) es maliciosa. Por ejemplo, la secuencia de inyección de código seguida de una escritura en disco puede considerarse un indicador de malicia. Las firmas de comportamiento de *malware* pueden permitir una alta tasa de detección, pero en general también producen una tasa relativamente alta de falsos positivos (entidades benignas etiquetadas falsamente como maliciosas). La reducción de la tasa de falsos positivos puede requerir aumentar la complejidad de las firmas de comportamiento de *malware*, lo cual puede aumentar sustancialmente la tara computacional.

A diferencia de dichos métodos basados en firmas, la presente invención introduce un conjunto de excepciones de regla para sustituir o complementar las firmas de comportamiento de *malware*. En algunas realizaciones, una excepción de regla comprende un conjunto de condiciones que, cuando son satisfechas por una tupla <evento, entidad desencadenante>, establecen que la entidad desencadenante respectiva es benigna (no maliciosa). En la presente, se dice que una tupla <evento, entidad desencadenante> que satisface las condiciones de una excepción de regla coincide con la excepción de regla respectiva. Un escenario de uso típico para tales excepciones de reglas comprende primero aplicar una firma de comportamiento de *malware* a un evento detectado. Cuando el evento coincide con una firma que indica malicia, la CSA **40** puede intentar además encontrar una coincidencia del evento con un conjunto de excepciones de regla. Cuando ninguna excepción de regla coincide con el evento detectado, la CSA **40** puede concluir que la entidad desencadenante es de hecho maliciosa. En cambio, cuando el evento coincide con al menos una excepción de regla, la CSA **40** puede concluir que la entidad desencadenante es benigna. Esta estrategia de análisis puede reducir sustancialmente la tasa de falsos positivos, al mismo tiempo que mantiene también bajo control la tara computacional. La reducción de los costes computacionales puede provenir, por ejemplo, del uso de firmas de comportamiento de *malware* más simples que en los sistemas convencionales de seguridad informática.

Los métodos convencionales de seguridad basados en el comportamiento incluyen suspender la ejecución de la entidad desencadenante mientras se analiza el evento de detección. Este tipo de análisis de eventos se conoce comúnmente como síncrono. Por el contrario, en algunas realizaciones de la presente invención, solo una parte del análisis de eventos se realiza síncronamente, mientras que otra parte del análisis se realiza de forma asíncrona. El término asíncrono en el presente documento se refiere a una manera de analizar un evento y/o excepción de regla, en donde la entidad desencadenante puede continuar con la ejecución, mientras que los datos sobre el evento/excepción respectivo se guardan para un análisis posterior.

En particular, en algunas realizaciones de la presente invención, algunas coincidencias de excepciones de regla se encuentran de forma síncrona, mientras que otras coincidencias de excepciones de regla se encuentran de forma asíncrona. El analizador **44** de excepciones síncronas puede configurarse para llevar a cabo un análisis síncrono de un evento que se produce dentro del sistema anfitrión o VM invitada protegido, para determinar si el mismo satisface (coincide con) cualquiera de un conjunto predeterminado de excepciones de regla. El conjunto de excepciones de regla verificadas por el analizador **44** de excepciones síncronas se considera en la presente como excepciones síncronas (más detalles a continuación). Las excepciones síncronas típicamente comprenden excepciones de regla cuya comparación con eventos requiere un coste computacional relativamente bajo. En algunas realizaciones, las excepciones síncronas pueden comprender un subconjunto de excepciones de regla que son críticas para evaluar el riesgo de seguridad que plantea una entidad monitorizada. En otro ejemplo, las excepciones síncronas incluyen excepciones de regla cuya verificación se basa exclusivamente en recursos locales para el sistema anfitrión **10** (por ejemplo, bases de datos de firmas almacenadas localmente en dispositivos **20** de almacenamiento).

A su vez, el analizador **46** de excepciones asíncronas puede configurarse para realizar un análisis asíncrono de un evento que se produce dentro de un sistema anfitrión o VM invitada protegido, para determinar si el mismo coincide con cualquiera de otro conjunto predeterminado de excepciones de regla. El conjunto de excepciones de regla verificadas por el analizador **46** de excepciones asíncronas se denominan en la presente excepciones asíncronas. A diferencia del análisis síncrono, el funcionamiento del módulo **46** de análisis asíncrono no está vinculado a la ejecución

de la entidad desencadenante, en el sentido de que la entidad desencadenante puede continuar con la ejecución, mientras que las operaciones que comparan el evento activado con excepciones pueden realizarse más adelante. Las excepciones asíncronas típicamente comprenden excepciones de regla cuya comparación con eventos requiere un coste computacional relativamente alto, o excepciones de regla que no se consideran críticas para la seguridad del sistema anfitrión **10.** Las operaciones ejemplificativas que forman parte de la comparación de excepciones asíncronas incluyen, entre otras, determinar la integridad de la entidad desencadenante (por ejemplo, utilizando comparación de valores *hash*), realizar una exploración remota de la entidad desencadenante (por ejemplo, enviando información sobre la entidad respectiva a un servidor remoto de la nube y recibiendo un veredicto de malicia del servidor respectivo) y determinar si la entidad desencadenante es el destinatario de código inyectado por otra entidad.

Las Figs.**4-A-B-C** muestran varias ubicaciones ejemplificativas del administrador **42** de notificaciones, del analizador **44** de excepciones síncronas y del analizador **46** de excepciones asíncronas de acuerdo con algunas realizaciones de la presente invención. Un profesional experto apreciará que los componentes **42-44-46** pueden ejecutarse fuera de una VM protegida (por ejemplo, en el nivel de privilegio de procesador del hipervisor **30**), dentro de una VM protegida (por ejemplo, en modo *kernel*) o dentro de una VM de seguridad aparte. Colocar un componente dentro de una VM protegida puede dar acceso al componente respectivo a una cantidad sustancial de información sobre entidades que se ejecutan dentro de la VM respectiva, pero puede hacer que el componente respectivo sea vulnerable al ataque de *software* malicioso que se ejecuta al mismo nivel de privilegio de procesador. Se podría usar un conjunto de técnicas, tales como alternar múltiples vistas EPT basadas en #VE (Excepción de Virtualización) y VMFUNC en plataformas Intel®, para mejorar la seguridad de componentes de seguridad ubicados dentro de una VM protegida. Cuando el componente respectivo se ejecuta fuera de la VM protegida, es relativamente seguro, pero ciertas operaciones que requieren desenredar la semántica de entidades y eventos pueden requerir un cálculo sustancial.

La Fig. **4-A** muestra una configuración ejemplificativa en la que el analizador **44** de excepciones síncronas se ejecuta fuera de una VM invitada protegida, mientras que en las Figs. **4-B-C**, el analizador **44** se ejecuta dentro de la VM invitada respectiva, en modo *kernel*. Teniendo en cuenta que el procesado síncrono suspende la ejecución de la entidad desencadenante y, por lo tanto, debe ser lo más rápido posible para no afectar a la experiencia del usuario, pueden ser preferibles configuraciones en las que el funcionamiento del analizador **44** de excepciones síncronas no requiere una salida costosa de la VM monitorizada. En una realización preferida, el analizador **44** de excepciones síncronas puede ejecutarse en el contexto del administrador **42** de notificaciones (por ejemplo, la Fig. **4-C**)**.**

El analizador **46** de excepciones asíncronas puede ejecutarse o bien fuera o bien dentro de una VM invitada protegida. En algunas realizaciones (véase, por ejemplo, la Fig. **4-C**), el analizador **46** puede ejecutarse en una VM **33** de seguridad dedicada expuesta en el sistema anfitrión, siendo la VM **33** de seguridad distinta de las VMs invitadas protegidas.

Dependiendo del contexto de ejecución de los componentes **42-44-46**, su funcionamiento puede requerir una señalización/mensajería compleja, a veces cruzando límites de máquinas virtuales. Dicha señalización puede llevarse a cabo utilizando cualquier planteamiento conocido en la técnica de virtualización de *hardware*. Por ejemplo, los datos pueden transmitirse a través de una sección de memoria compartida entre dos componentes, y la señalización puede comprender una combinación de salidas de VM e inyecciones de eventos.

La Fig. **5** muestra un intercambio ejemplificativo entre componentes **42**, **44**, **46** y **48** de acuerdo con algunas realizaciones de la presente invención. Cuando una entidad desencadenante provoca la manifestación de un evento dentro de una VM invitada, el evento provoca la entrega de una notificación **52** de evento (p. ej., evento de procesador, tal como salida de VM o excepción de virtualización) al administrador **42.** El administrador **42** a continuación puede determinar un tipo y un conjunto de parámetros del evento notificado en ese momento. Los tipos de evento ejemplificativos incluyen, entre otros, la inyección de código, una llamada de sistema particular, la creación de un archivo de disco y una solicitud HTTP. Los parámetros del evento pueden ser específicos de cada tipo de evento notificado. Algunos parámetros de evento ejemplificativos incluyen, entre otros, un identificador de un proceso o hilo (por ejemplo, ID de proceso) que provoca el evento notificado, un nombre de archivo, una ruta, una dirección de memoria y un operando de una instrucción de procesador.

En algunas realizaciones, el administrador **42** a continuación pasa un indicador **54** de evento al analizador **44** de excepciones síncronas**.** El indicador **54** de evento puede comprender, entre otros, un identificador exclusivo del evento respectivo (ID de evento), un indicador de un tipo de evento y un conjunto de parámetros de evento. El analizador **44** de excepciones síncronas a continuación puede intentar encontrar una coincidencia del evento respectivo con un conjunto de excepciones de regla síncronas, por ejemplo, consultando una base **50** de conocimiento de excepciones.

En algunas realizaciones, la base **50** de conocimiento de excepciones almacena un conjunto de entradas de excepción de regla, por ejemplo, en soportes legibles por ordenador que forman parte del sistema anfitrión **10** ó están acoplados comunicativamente al mismo**.** La Fig. **6** muestra un formato ejemplificativo de una entrada **60** de excepción de regla de acuerdo con algunas realizaciones de la presente invención. La entrada **60** de excepción incluye una bandera **62** de sincronía, que puede indicar si la entrada respectiva tiene una parte asíncrona o no. La entrada **60** puede comprender además un indicador **64** de tipo de evento indicativo de un tipo de evento (por ejemplo, un ID numérico asociado de forma exclusiva a cada tipo de evento). El indicador **64** de tipo de evento puede permitir a la base **50** de conocimiento recuperar selectivamente una entrada de excepción de regla de acuerdo con un tipo de evento

detectado. Alternativamente, la base **50** de conocimiento puede mantener una asignación interna (por ejemplo, índice *hash*) que asocia cada entrada de excepción de regla con un tipo de evento relevante para la excepción de regla respectiva.

La entrada **60** de excepción de regla puede comprender además una firma **66** de excepción síncrona y una firma **68** de excepción asíncrona**.** La firma **66** comprende una codificación de una excepción de regla síncrona, es decir, una codificación de un conjunto de condiciones que el analizador **46** debe verificar síncronamente**.** Por el contrario, la firma asíncrona **68** comprende una codificación de una excepción de regla asíncrona, es decir, una codificación de un conjunto de condiciones que se verificará asíncronamente por parte de un analizador **46** de excepciones asíncronas**.**

Cuando una entrada de excepción de regla relevante para el tipo en curso de evento notificado tiene una parte asíncrona, el analizador **44** de excepciones síncronas puede insertar una solicitud de análisis de excepción (EAR) **56** en una lista **58** de EAR para su procesado posterior (más detalles posteriormente). En algunas realizaciones, la lista **58** de EAR comprende una estructura de datos que tiene una pluralidad de entradas, y cada entrada codifica una solicitud de comparación de excepción de regla asíncrona. La lista **58** de EAR puede organizarse como una cola de múltiples productores y múltiples consumidores (por ejemplo, una cola del tipo primero en entrar, primero en salir). La Fig. **7** proporciona un formato ejemplificativo de una solicitud de análisis de excepción de acuerdo con algunas realizaciones de la presente invención. La EAR ilustrada comprende un ID de evento asociado de manera exclusiva al evento que activó el análisis respectivo. La EAR **56** puede incluir además un ID de excepción que identifique de forma exclusiva una entrada particular **60** de la base **50** de conocimiento de excepciones. La EAR **56** puede comprender además un indicador de la entidad desencadenante. La inclusión de un ID de entidad de este tipo en la EAR **56** puede ayudar al supervisor **48** de terminación a determinar si hay solicitudes de análisis de excepciones pendientes asociadas con una entidad ejecutora particular (véanse más detalles a continuación). En algunas realizaciones, una EAR **56** incluye además varios datos de contexto determinados por el administrador **42** de notificaciones y/o el analizador **44** de excepciones síncronas, datos de contexto que comprenden información sobre el evento respectivo y/o sobre la entidad desencadenante. Los datos de contexto pueden incluir, entre otros, direcciones de memoria, un ID de proceso, un valor del puntero de instrucción (RIP) correspondiente al momento en que se generó el evento activado, etc. Tales datos de contexto pueden ser utilizados por un analizador **46** de excepciones asíncronas cuando se realiza una comparación de excepción de regla asíncrona.

La Fig. **8** muestra una secuencia ejemplificativa de pasos realizados por el administrador **42** de notificaciones de acuerdo con algunas realizaciones de la presente invención. Como se ha mostrado anteriormente, el administrador **42** se ejecuta en una posición que permite al administrador **42** suspender efectivamente la ejecución de la entidad desencadenante. Por ejemplo, el administrador **42** puede registrarse como administrador de eventos para salidas de VM y/o excepciones de virtualización, en donde dichos eventos del procesador se activan en respuesta a la manifestación de un evento monitorizado específico provocado por *software* (por ejemplo, un intento de acceder a una página de memoria en particular). En respuesta a tales eventos, el procesador **12** suspende la ejecución de la entidad desencadenante y cambia a ejecutar el administrador **42** de notificaciones**.** Al controlador **42** de notificaciones por lo tanto, se le notifica la manifestación del evento monitorizado, mientras se suspende la ejecución de la entidad desencadenante. Cuando el evento del procesador es una salida de VM, el administrador **42** puede ejecutarse en el nivel de hipervisor **30.**

Cuando el administrador **42** recibe una notificación de evento, una secuencia de pasos **106-108** determina un tipo de evento que se produjo dentro de la VM invitada protegida y un conjunto de parámetros del evento respectivo. El administrador **42** a continuación puede transmitir el indicador **54** de evento al analizador **44** de excepciones síncronas (paso **110**)**.** En un paso **112**, el administrador **42** puede esperar una señal de liberación del analizador **44.** En algunas realizaciones, dicha señal de liberación indica o bien que el evento en curso ha coincidido con una excepción de regla síncrona o bien que una EAR asociada al evento en curso se añadió a la lista **58** de EAR (véanse detalles a continuación, en relación con la Fig. **9**)**.** En respuesta a la recepción de la señal de liberación, el administrador **42** puede ordenar al procesador **12** que reanude la ejecución de la entidad desencadenante.

La Fig. **9** muestra una secuencia ejemplificativa de pasos realizados por el analizador **44** de excepciones síncronas de acuerdo con algunas realizaciones de la presente invención. En respuesta a la recepción del indicador **54** de evento desde el administrador **42** de notificaciones, un paso **124** realiza una evaluación preliminar de seguridad de la entidad desencadenante. Por ejemplo, el analizador **44** puede determinar si el evento respectivo es indicativo de *malware*. En algunas realizaciones, el paso **124** intenta encontrar una coincidencia de la tupla en curso <evento, entidad> con un conjunto de firmas de comportamiento de *malware*. Un ejemplo de tales firmas indicativas de *malware* comprende la secuencia de eventos: una primera entidad descarga un archivo ejecutable sin una firma digital válida, la primera entidad lanza una segunda entidad desde el archivo ejecutable y la segunda entidad intenta registrarse para un inicio automático en el arranque del sistema. Otro ejemplo de firma de comportamiento de *malware* comprende un controlador que intenta sobrescribir una entrada de la Tabla de Descriptores de Servicio del Sistema (SSDT). Cuando la evaluación preliminar indica que la entidad desencadenante probablemente no sea maliciosa, el analizador **44** puede indicar al administrador **42** de notificaciones que reanude la ejecución de la entidad desencadenante (véase anteriormente).

Cuando la evaluación preliminar indica que la entidad desencadenante es sospechosa de malicia, un paso **126** recupera selectivamente un conjunto de entradas de excepción de regla de la base **50** de conocimiento de excepciones

según el tipo de evento correspondiente al evento en curso. Cuando la base **50** de conocimiento no contiene ninguna entrada de excepción de regla asociada al tipo del evento en curso, en un paso **130** la CSA **40** puede realizar una acción anti-*malware* contra la entidad desencadenante. Dicha acción protectora puede incluir, entre otros, terminar, poner en cuarentena o incapacitar de otra manera a la entidad desencadenante, y revertir un conjunto de cambios provocados sobre el sistema anfitrión **10** como resultado de la ejecución de la entidad desencadenante. En algunas realizaciones, la CSA **40** mantiene un conjunto de puntuaciones indicativas de *malware* asociadas a cada entidad monitorizada. El paso **130** puede comprender incrementar la(s) puntuación(es) respectiva(s) en una cantidad que puede ser específica de cada evento. La CSA **40** puede comparar además la(s) puntuación(es) con un umbral y llevar a cabo una acción anti-*malware* solo cuando, por ejemplo, una puntuación supere un umbral predeterminado.

Cuando la base **50** de conocimiento contiene al menos una entrada **60** de excepción de regla asociada al tipo de evento correspondiente al evento en curso, un paso **131** determina si el evento en curso coincide con una excepción de regla síncrona de la entrada de excepción de regla respectiva. El paso **131** puede comprender probar si un conjunto de condiciones codificadas por la firma síncrona **66** de la excepción de regla respectiva son satisfechas por la tupla <evento en curso, entidad desencadenante>. Por ello, el paso **131** puede incluir llevar a cabo un conjunto de cálculos, por ejemplo para determinar un tipo de entidad de la entidad desencadenante de acuerdo con el indicador **54** de evento**.** Los tipos de entidad ejemplificativos incluyen componentes específicos del OS **34**, una instancia de una aplicación en particular (por ejemplo, Adobe® Acrobat Reader®, Microsoft® Word®), una categoría particular de entidades (por ejemplo, administrador de archivos, navegador), etc. Otros ejemplos de tipos de entidad incluyen un controlador, una biblioteca compartida (por ejemplo, una biblioteca de enlace dinámico - DLL) y una sección de código inyectada.

Cuando no se encuentra ninguna coincidencia de la firma síncrona, el analizador **44** concluye que la entidad desencadenante es realmente maliciosa y avanza al paso **130** descrito anteriormente. Cuando el evento en curso coincide con la firma síncrona de al menos una excepción de regla, en un paso **134,** el analizador **44** determina si la entrada de excepción de regla respectiva también comprende una firma asíncrona. En algunas realizaciones, el paso **134** incluye verificar el valor de la bandera **62** de sincronía (véase la Fig. **6**)**.** Cuando sea negativo, el analizador **44** señaliza al administrador **42** que reanude la ejecución de la entidad desencadenante. Cuando sea afirmativo, un paso **136** determina datos de contexto sobre el evento en curso y/o sobre la entidad desencadenante. Un paso más **138** formula una EAR **56** y añade la EAR **56** a la lista **58** de EAR**.** El analizador **44** de excepciones síncronas entonces puede indicar al administrador **42** que reanude la ejecución de la entidad desencadenante.

La Fig. **10** muestra una secuencia ejemplificativa de pasos realizados por una instancia del tipo mencionado del analizador **46** de excepciones asíncronas**.** En algunas realizaciones, la aplicación **40** de seguridad informática administra un grupo de hilos para la comparación de excepciones de regla asíncronas. Los hilos del grupo pueden ejecutarse fuera o dentro de una VM protegida, o dentro de una VM de seguridad aparte (véase, por ejemplo, la Fig. **4-C**)**.** Cuando dichos hilos se ejecutan dentro de la VM protegida respectiva, pueden ejecutarse en modo *kernel* (anillo 0). Cada vez que queda disponible un hilo del grupo, la CSA **40** puede lanzar una instancia del analizador **46** de excepciones asíncronas**.**

En una secuencia de pasos **142-144**, el analizador **46** de excepciones asíncronas determina si en ese momento hay solicitudes de análisis de excepciones pendientes. Cuando la lista **58** de EAR no está vacía, el analizador **46** puede eliminar una EAR de la lista **58** e intentar encontrar una coincidencia del evento indicado por la EAR respectiva con una excepción de regla asíncrona indicada por la EAR respectiva. El paso **148** puede comprender una determinación adicional de parámetros del evento y/o de información sobre la entidad que activa el evento respectivo. Dichos cálculos pueden incluir, por ejemplo, el cálculo de valores *hash*, búsquedas en memoria, el establecimiento de relaciones de filiación entre varias entidades de *software* (por ejemplo, qué procesos han generado otros procesos), una emulación, etc. En algunas realizaciones, el paso **148** incluye el intercambio de datos con un servidor de seguridad remoto (exploración en la nube).

En algunas realizaciones, cuando el evento respectivo coincide con la excepción de regla asíncrona, el analizador **46** finaliza. La terminación de la instancia en curso del analizador **46** indica que el evento respectivo no es indicativo de una amenaza a la seguridad informática y, por lo tanto, no es necesario realizar más análisis del evento respectivo. Cuando el analizador **46** determina que el evento no coincide con la excepción de regla asíncrona respectiva, un paso **152** puede indicar a la CSA **40** que lleve a cabo una acción de protección anti-*malware* (véase anteriormente, paso **130** en la Fig. **9**).

En una realización alternativa, la excepción de regla asíncrona se formula de modo que una coincidencia indica que la entidad respectiva es maliciosa. En tales realizaciones, el analizador **46** puede finalizar cuando el evento no coincide con la excepción de regla respectiva y la CSA **40** puede llevar a cabo una acción anti-*malware* cuando el evento coincide con la excepción de regla respectiva.

La Fig. **11** muestra una secuencia ejemplificativa de pasos realizados por el supervisor **48** de terminación de acuerdo con algunas realizaciones de la presente invención. El supervisor **48** de terminación puede ejecutarse fuera o dentro de la máquina virtual protegida, y está acoplado comunicativamente al menos con el analizador **46** de excepciones asíncronas**.**

Dado que el análisis de excepciones de regla asíncronas no está vinculado a la ejecución de la entidad desencadenante, puede surgir una situación en la que la entidad desencadenante termine la ejecución antes de que el análisis **46** de excepciones asíncronas consiga procesar una solicitud de análisis de excepción relacionada con la entidad respectiva. En tales situaciones, una entidad maliciosa puede escapar sin ser detectada, o puede hacer algún daño que no pueda revertirse. Para evitar tales situaciones, en algunas realizaciones de la presente invención, el supervisor **48** de terminación detecta un intento por parte del OS de terminar una entidad (pasos **162-164**)**.** La detección del intento de terminación puede suspender efectivamente la ejecución de la entidad de terminación. El paso **162** puede comprender colaborar con el administrador **42** de notificaciones**,** por ejemplo, el administrador **42** puede detectar en realidad el intento de terminación y señalizarlo al supervisor **48** de terminación**.**

En respuesta a la detección de un intento de terminación, en una secuencia de pasos **166-168**, el supervisor **48** de terminación puede determinar si todavía hay solicitudes de análisis de excepción pendientes para la entidad de terminación. Cuando no las haya, un paso **170** puede ordenar al procesador **12** que reanude la ejecución de la entidad que termina, permitiendo en efecto que la entidad respectiva termine. En algunas realizaciones, el paso **170** comprende enviar una señal de liberación al administrador **42** de notificaciones, ordenando al administrador **42** que libere la entidad de terminación.

Cuando la lista **58** de EAR contiene al menos una EAR asociada a la entidad de terminación, el supervisor **48** de terminación puede mantener la entidad respectiva suspendida hasta que se procesen todas estas solicitudes pendientes. Una secuencia de pasos **172-174** fuerza el procesado de una solicitud asociada a la entidad de terminación respectiva (la secuencia puede repetirse hasta que se procesen todas estas solicitudes). El paso **174** puede incluir invocar al analizador **46** de excepciones asíncronas para procesar cada EAR pendiente relacionada con la entidad de terminación. Cuando el analizador **46** determina que el evento indicado por la EAR respectiva no coincide con la excepción de regla asíncrona indicada por la EAR respectiva, un paso **178** puede indicar a la CSA **40** que lleve a cabo una acción de protección contra la entidad de terminación (véase anteriormente, en relación con las Figs. **9-10**).

Los sistemas y métodos ejemplificativos descritos anteriormente permiten una monitorización eficiente del comportamiento de entidades de *software*. En algunas realizaciones, se despliega un mecanismo de notificación para detectar la manifestación de eventos relevantes para la seguridad dentro de un sistema anfitrión o máquina virtual protegido, y para informar de los eventos respectivos a *software* de seguridad. A continuación, el *software* de seguridad analiza los eventos respectivos para determinar si son indicativos de una amenaza a la seguridad informática, tal como *malware*, *software* espía, una intrusión no autorizada, etc.

Algunos sistemas y métodos convencionales basados en el comportamiento se basan en firmas de comportamiento de *malware* para determinar si una entidad es maliciosa. Una firma de comportamiento de *malware* comprende típicamente un conjunto de condiciones que, cuando son satisfechas por una tupla <evento, entidad>, establecen que el evento respectivo es indicativo de malicia y, por lo tanto, que es probable que la entidad de *software* que activa el evento respectivo sea maliciosa. Para evitar que la entidad desencadenante realice sus actividades maliciosas, los sistemas y métodos convencionales suspenden la ejecución de la entidad desencadenante mientras el evento activado se analiza en busca de indicadores de malicia.

Algunas realizaciones de la presente invención se basan en dos observaciones. Primero, no todas las manifestaciones de un tipo particular de evento son igualmente indicativas de *malware*. El mismo tipo de evento (por ejemplo, acceder a un URL, abrir un archivo de disco, etc.) puede indicar malicia en algunos escenarios, mientras que puede ser completamente benigno en otros escenarios. En uno de esos ejemplos, un evento puede no ser indicativo de malicia cuando se considera de forma aislada, pero puede ser indicativo de *malware* cuando ocurre como parte de una secuencia específica de eventos. Por ejemplo, escribir en un archivo de disco puede ser una operación benigna cuando se considera de forma aislada (es decir, muchos procesos y aplicaciones acceden legítimamente al disco). Sin embargo, el evento de escritura puede ser sospechoso cuando la entidad que realiza la escritura es el destinatario de código inyectado desde otra entidad. Esta observación sugiere que la detección exitosa de *malware* puede requerir firmas de comportamiento de *malware* bastante complejas, que puedan discernir entre varios escenarios como se ha descrito anteriormente. El uso de firmas de comportamiento tan complejas conlleva típicamente un coste computacional relativamente alto. Además, la optimización de firmas de comportamiento de *malware* para obtener una alta tasa de detección conduce típicamente a un aumento de detecciones de falsos positivos (eventos benignos clasificados erróneamente como indicativos de *malware*, entidades legítimas clasificadas erróneamente como maliciosas). La clasificación de falsos positivos es particularmente rechazable en el campo de la seguridad informática, ya que puede derivar en la pérdida de datos y la pérdida de productividad para el usuario.

La segunda observación es que suspender la ejecución de la entidad desencadenante durante todo el tiempo correspondiente a la comparación de las firmas de comportamiento de *malware* afecta negativamente a la experiencia del usuario. Esto es especialmente cierto en el caso de firmas de comportamiento complejas, y en configuraciones de virtualización de *hardware* en las que el análisis de eventos se realiza desde una posición fuera de la VM donde se ha producido el evento (por ejemplo, desde el nivel del hipervisor **30**).

A diferencia de tales sistemas y métodos convencionales de seguridad informática, algunas realizaciones usan un conjunto de firmas de excepción de regla para complementar firmas de comportamiento de *malware*. Una firma de excepción de regla comprende un conjunto de condiciones que, cuando son satisfechas por una tupla <evento,

entidad>, establecen que el evento respectivo es benigno y, por lo tanto, que la entidad desencadenante no es maliciosa. Por lo tanto, una excepción de regla proporciona una codificación de una excepción a una regla que normalmente indicaría un comportamiento malicioso. En un escenario de caso de uso ejemplificativo, el *software* de seguridad puede en primer lugar intentar encontrar una coincidencia de un evento detectado con un conjunto de firmas de comportamiento de *malware* relativamente simples y computacionalmente económicas, para determinar si es probable que la entidad desencadenante sea maliciosa. En caso afirmativo, el *software* de seguridad puede intentar además encontrar una coincidencia del evento respectivo con un conjunto de firmas de excepción de regla. Una coincidencia de una firma de excepción de regla puede indicar que la entidad desencadenante es realmente benigna.

La adición de firmas de excepción de regla crea la oportunidad de usar firmas relativamente simples en lugar de las firmas bastante complejas necesarias cuando la coincidencia de firmas de comportamiento de *malware* se usa de manera individual. Por lo tanto, algunas realizaciones reducen la tara computacional producida por el *software* de seguridad, al tiempo que reducen también la tasa de detecciones de falsos positivos.

Además, en algunas realizaciones de la presente invención, la comparación de firmas de excepción de regla se realiza al menos en parte de una manera asíncrona, es decir, mientras la entidad que activó el evento respectivo puede continuar con la ejecución. Al elegir no suspender la ejecución de la entidad desencadenante durante todo el tiempo correspondiente al análisis de seguridad, el impacto sobre la experiencia del usuario se reduce significativamente. En algunas realizaciones, las firmas de excepción de regla se optimizan para obtener una tara baja: las firmas que conllevan un coste computacional relativamente bajo se usan en la comparación síncrona, mientras que las firmas que son relativamente costosas se usan en la comparación asíncrona.

Si bien existe cierta similitud entre las firmas convencionales de comportamiento de *malware* y las firmas de excepción de regla, su uso y semántica son bastante diferentes. Por ejemplo, la afirmación de que las firmas de excepción de reglas son simplemente el complemento o lo contrario de las firmas de comportamiento de *malware* no es cierta. Las firmas de comportamiento de *malware* y las firmas de excepción de regla no son mutuamente excluyentes. Por ejemplo, cuando un evento coincide con una firma de comportamiento de *malware*, no significa que no pueda coincidir también con una firma de excepción de regla. En cambio, es precisamente en situaciones en las que un evento coincide tanto con firmas de comportamiento de *malware* como con firmas de excepción de regla donde las firmas de excepción de regla son más valiosas, ya que permiten un proceso de decisión eficiente de detección de *malware*.

Un ejemplo de excepciones de regla síncronas frente a asíncronas y de su relación con firmas de comportamiento de *malware* comprende la detección de un intento, por parte de una entidad de *software* monitorizada, de parchear el código de un módulo ejecutable compartido (por ejemplo, biblioteca). El parcheo de código suele indicar malicia, por lo que puede codificarse como tal en una firma de comportamiento de *malware*. El uso de la firma respectiva puede activar una alerta de *malware* cada vez que se detecta un parcheo de código. Sin embargo, varias entidades de *software* (por ejemplo, el OS) aplican parches de código legítimos, por ejemplo, al lanzar un proceso nuevo. Del mismo modo, un proceso de un conjunto de aplicaciones de *software* determinado (por ejemplo, Microsoft® Office®) puede parchear legítimamente otro proceso del mismo conjunto. En algunas realizaciones de la presente invención, tales situaciones pueden abordarse usando excepciones de reglas. Una excepción de regla síncrona ejemplificativa puede verificar si el proceso de parcheo es uno de los procesos fiables del OS y si el proceso de destino (el parcheado) está en el arranque. Cuando se cumplen ambas condiciones, entonces la entidad que lleva a cabo el parcheo se considera benigna (legítima). De esta manera, cuando el OS inicia un proceso y aplica un parcheo, puede continuar sin ser bloqueado por la CSA **40**. Por el contrario, según la excepción de regla anterior, si un proceso desconocido intenta realizar la misma operación de parcheo, se bloqueará. Para permitir que algunos procesos desconocidos apliquen parches, algunas realizaciones pueden usar una firma de excepción de regla de dos partes: la parte síncrona puede verificar la identidad de las entidades que participan en el parcheo de código, mientras que la parte asíncrona puede verificar la propia memoria intermedia/código inyectado (p. ej. desensamblándolo y/o buscando en él un patrón de código específico). Cuando el código inyectado no es malicioso, el proceso desconocido puede considerarse benigno.

En otro ejemplo, una firma de comportamiento de *malware* puede indicar que la inyección de código es indicativa de *malware*. Una firma de excepción de regla síncrona puede permitir la inyección de código cuando el proceso que realiza la inyección es bien conocido y fiable. Sin embargo, la misma firma de excepción de regla puede tener una parte asíncrona, que realiza el análisis de contenido del código inyectado. Cuando el contenido parece inusual para ese proceso en particular, el proceso respectivo puede considerarse malicioso. En este ejemplo, el análisis de contenido, que es relativamente costoso en términos de tara computacional, se realiza de forma asíncrona, es decir, mientras se ejecuta el proceso respectivo, para tener un efecto mínimo sobre la experiencia del usuario.

En otro ejemplo más, una firma de comportamiento de *malware* puede indicar que un intento de un navegador para cargar un complemento es indicativo de *malware*. Sin embargo, algunos complementos son benignos y se les debe permitir funcionar. En una realización ejemplificativa, una excepción de regla síncrona puede comprobar si el complemento respectivo está firmado digitalmente por una determinada autoridad y, en caso afirmativo, determinar que el navegador es benigno. En el caso de una comparación de firma síncrona, se puede permitir que el navegador cargue y ejecute el complemento. Una excepción de regla adicional puede determinar, a continuación, asíncronamente si el certificado utilizado para firmar el complemento es válido en ese momento o ha sido revocado. Cuando el certificado respectivo ha sido revocado, la CSA **40** puede terminar el navegador y/o mostrar una alerta. La comprobación de la validez del certificado requiere típicamente el envío de una solicitud a un servidor remoto y, por lo

tanto, puede afectar sustancialmente a la experiencia del usuario si se realiza de forma síncrona.

Para un profesional experto en la materia resultará evidente que las realizaciones anteriores pueden modificarse de muchas maneras sin desviarse del alcance de la invención. Por consiguiente, el alcance de la invención debe determinarse por las siguientes reivindicaciones y sus equivalentes legales.

5

REIVINDICACIONES

1. Sistema anfitrión que comprende un procesador de *hardware* y una memoria, estando configurado el procesador de *hardware* para ejecutar una entidad objetivo, un analizador de excepciones síncronas y un analizador de excepciones asíncronas, en el que el procesador de *hardware* está configurado además para:

en respuesta a la detección de una manifestación de un evento provocado por una ejecución de la entidad objetivo, suspender la ejecución de la entidad objetivo, y

en respuesta a la suspensión de la ejecución de la entidad objetivo, cambiar a la ejecución del analizador de excepciones síncronas;

en donde el analizador de excepciones síncronas está configurado para:

determinar si la entidad objetivo es sospechosa de ser maliciosa según el evento,

en respuesta, cuando la entidad objetivo es sospechosa de ser maliciosa, recuperar selectivamente una firma de excepción de regla de una pluralidad de firmas de excepción de regla, recuperándose la firma de excepción de regla de acuerdo con el evento, en donde la firma de excepción de regla comprende una codificación de una primera condición y una codificación de una segunda condición,

en respuesta a la recuperación de la firma de excepción de regla, determinar si se cumple la primera condición según el evento y según la entidad objetivo,

en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, provocar que el procesador de *hardware* reanude la ejecución de la entidad objetivo, y

en respuesta a la determinación de si se cumple la primera condición, cuando la primera condición no se cumple, determinar que la entidad objetivo es maliciosa; y

caracterizado porque

estando configurado el analizador de excepciones asíncronas para:

en respuesta a que el procesador de *hardware* reanude la ejecución de la entidad objetivo, determinar si se cumple la segunda condición de acuerdo con el evento y de acuerdo con la entidad objetivo,

en respuesta a la determinación de si se cumple la segunda condición, cuando se cumple la segunda condición, determinar que la entidad objetivo no es maliciosa, y

en respuesta a la determinación de si se cumple la segunda condición, cuando la segunda condición no se cumple, determinar que la entidad objetivo es maliciosa.

2. Sistema anfitrión de la reivindicación 1, en el que:

el analizador de excepciones síncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, para insertar una solicitud de análisis en una cola de solicitudes, formulándose la solicitud de análisis de acuerdo con la segunda condición, de acuerdo con el evento, y además de acuerdo con la entidad objetivo; y

el analizador de excepciones asíncronas está configurado, además, como preparación para determinar si se cumple la segunda condición, para eliminar la solicitud de análisis de la cola de solicitudes.

3. Sistema anfitrión de la reivindicación 2, en el que el procesador de *hardware* está configurado además para:

en respuesta a la detección de un intento de terminar la entidad objetivo, suspender el intento; y

en respuesta a la suspensión del intento, cambiar a la ejecución de un supervisor de terminación conectado al analizador de excepciones asíncronas, estando configurado el supervisor de terminación para:

buscar en la cola de solicitudes una segunda solicitud de análisis formulada de acuerdo con la entidad objetivo, indicando la segunda solicitud de análisis una tercera condición,

invocar al analizador de excepciones asíncronas para procesar la segunda solicitud de análisis, y

en respuesta a la invocación del analizador de excepciones asíncronas, cuando el analizador de excepciones asíncronas determina que se cumple la tercera condición, provocar que el procesador de *hardware* reanude el intento de terminar la entidad objetivo.

4. Sistema anfitrión de la reivindicación 1, en el que la entidad objetivo se ejecuta dentro de una máquina virtual

invitada expuesta por el sistema anfitrión, y en el que el analizador de excepciones asíncronas se ejecuta fuera de la máquina virtual invitada.

5. Sistema anfitrión de la reivindicación 4, en el que el analizador de excepciones asíncronas se ejecuta dentro de una máquina virtual de seguridad expuesta por el sistema anfitrión, ejecutándose simultáneamente la máquina virtual de seguridad con la máquina virtual invitada.

6. Sistema anfitrión de la reivindicación 1, en el que la firma de excepción de regla está configurada de modo que la determinación de si se cumple la primera condición conlleva un coste computacional sustancialmente menor que la determinación de si se cumple la segunda condición.

7. Sistema anfitrión de la reivindicación 1, en el que el evento comprende un intento de acceder a la memoria de una manera que viola un permiso de acceso a la memoria.

8. Sistema anfitrión de la reivindicación 1, en el que:

determinar si se cumple la primera condición comprende determinar si la entidad objetivo ha inyectado código en una segunda entidad; y

determinar si se cumple la segunda condición comprende determinar si el código es malicioso.

9. Soporte no transitorio legible por ordenador que almacena instrucciones de procesador que, cuando se ejecutan por un procesador de *hardware* de un sistema anfitrión, provocan que el sistema anfitrión forme un analizador de excepciones síncronas y un analizador de excepciones asíncronas, en el que el procesador de *hardware* está configurado para:

en respuesta a la detección de una manifestación de un evento provocado por una ejecución de la entidad objetivo, suspender la ejecución de la entidad objetivo, y

en respuesta a la suspensión de la ejecución de la entidad objetivo, cambiar a la ejecución del analizador de excepciones síncronas;

en donde el analizador de excepciones síncronas está configurado para:

determinar si la entidad objetivo es sospechosa de malicia según el evento,

en respuesta, cuando la entidad objetivo es sospechosa de malicia, recuperar selectivamente una firma de excepción de regla de una pluralidad de firmas de excepción de regla, recuperándose la firma de excepción de regla de acuerdo con el evento, en donde la firma de excepción de regla comprende una codificación de una primera condición y una codificación de una segunda condición,

en respuesta a la recuperación de la firma de excepción de regla, determinar si se cumple la primera condición según el evento y según la entidad objetivo,

en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, provocar que el procesador de *hardware* reanude la ejecución de la entidad objetivo, y

en respuesta a la determinación de si se cumple la primera condición, cuando la primera condición no se cumple, determinar que la entidad objetivo es maliciosa; y

caracterizado porque

el analizador de excepciones asíncronas está configurado para:

en respuesta a que el procesador de *hardware* reanude la ejecución de la entidad objetivo, determinar si se cumple la segunda condición de acuerdo con el evento y de acuerdo con la entidad objetivo,

en respuesta a la determinación de si se cumple la segunda condición, cuando se cumple la segunda condición, determinar que la entidad objetivo no es maliciosa, y

en respuesta a la determinación de si se cumple la segunda condición, cuando la segunda condición no se cumple, determinar que la entidad objetivo es maliciosa.

10. Soporte legible por ordenador de la reivindicación 9, en el que:

el analizador de excepciones síncronas está configurado adicionalmente, en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, para insertar una solicitud de análisis en una cola de solicitudes, formulándose la solicitud de análisis de acuerdo con la segunda condición, de acuerdo con el evento, y además de acuerdo con la entidad objetivo; y

el analizador de excepciones asíncronas está configurado además, como preparación para determinar si se cumple la segunda condición, para eliminar la solicitud de análisis de la cola de solicitudes.

11. Soporte legible por ordenador de la reivindicación 10, en el que el procesador de *hardware* está configurado además para:

en respuesta a la detección de un intento de terminar la entidad objetivo, suspender el intento; y

en respuesta a la suspensión del intento, cambiar a la ejecución de un supervisor de terminación conectado al analizador de excepciones asíncronas, estando configurado el supervisor de terminación para:

buscar en la cola de solicitudes una segunda solicitud de análisis formulada de acuerdo con la entidad objetivo, indicando la segunda solicitud de análisis una tercera condición,

invocar al analizador de excepciones asíncronas para procesar la segunda solicitud de análisis, y

en respuesta a la invocación del analizador de excepciones asíncronas, cuando el analizador de excepciones asíncronas determina que se cumple la tercera condición, provocar que el procesador de *hardware* reanude el intento de terminar la entidad objetivo.

12. Soporte legible por ordenador de la reivindicación 9, en donde la entidad objetivo se ejecuta dentro de una máquina virtual invitada expuesta por el sistema anfitrión, y en donde el analizador de excepciones asíncronas se ejecuta fuera de la máquina virtual invitada.

13. Soporte legible por ordenador de la reivindicación 12, en el que el analizador de excepciones asíncronas se ejecuta dentro de una máquina virtual de seguridad expuesta por el sistema anfitrión, ejecutándose la máquina virtual de seguridad simultáneamente con la máquina virtual invitada.

14. Soporte legible por ordenador de la reivindicación 9, en el que la firma de excepción de regla está configurada de modo que determinar si se cumple la primera condición conlleva un coste computacional sustancialmente menor que determinar si se cumple la segunda condición.

15. Soporte legible por ordenador de la reivindicación 9, en el que el evento comprende un intento de acceder a una memoria del sistema anfitrión de una manera que viola un permiso de acceso a la memoria.

16. Soporte legible por ordenador de la reivindicación 9, en el que:

determinar si se cumple la primera condición comprende determinar si la entidad objetivo ha inyectado código en una segunda entidad; y

determinar si se cumple la segunda condición comprende determinar si el código es malicioso.

17. Método para proteger un sistema anfitrión contra amenazas de seguridad informática, sistema anfitrión que comprende un procesador de *hardware* y una memoria, comprendiendo el método:

utilizar el procesador de *hardware* para detectar una manifestación de un evento provocado por una ejecución de una entidad objetivo;

en respuesta a la detección de la manifestación del evento, utilizar el procesador de *hardware* para suspender la ejecución de la entidad objetivo;

en respuesta a la suspensión de la ejecución de la entidad objetivo, utilizar el procesador de *hardware* para cambiar a la ejecución de un analizador de excepciones síncronas configurado para:

determinar si la entidad objetivo es sospechosa de malicia según el evento,

en respuesta, cuando la entidad objetivo es sospechosa de malicia, recuperar selectivamente una firma de excepción de regla de una pluralidad de firmas de excepción de regla, recuperándose la firma de excepción de regla de acuerdo con el evento, en donde la firma de excepción de regla comprende una codificación de una primera condición y una codificación de una segunda condición,

en respuesta a la recuperación de la firma de excepción de regla, determinar si se cumple la primera condición según el evento y según la entidad objetivo,

en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, provocar que el procesador de *hardware* reanude la ejecución de la entidad objetivo, y

en respuesta a la determinación de si se cumple la primera condición, cuando la primera condición no se cumple, determinar que la entidad objetivo es maliciosa;

caracterizado porque

en respuesta a que el procesador de *hardware* reanude la ejecución de la entidad objetivo, utilizar el procesador de *hardware* para determinar si se cumple la segunda condición de acuerdo con el evento y de acuerdo con la entidad objetivo;

en respuesta a determinar si se cumple la segunda condición, cuando se cumple la segunda condición, determinar que la entidad objetivo no es maliciosa; y

en respuesta a determinar si se cumple la segunda condición, cuando la segunda condición no se cumple, determinar que la entidad objetivo es maliciosa.

18. Método de la reivindicación 17, que comprende, además:

en respuesta a la determinación de si se cumple la primera condición, cuando se cumple la primera condición, utilizar el procesador de *hardware* para insertar una solicitud de análisis en una cola de solicitudes, formulándose la solicitud de análisis de acuerdo con la segunda condición, de acuerdo con el evento, y además de acuerdo con la entidad objetivo; y

como preparación para determinar si se cumple la segunda condición, utilizar el procesador de *hardware* para eliminar la solicitud de análisis de la cola.

19. Método de la reivindicación 18, que comprende, además:

en respuesta a la detección de un intento de terminar la entidad objetivo, utilizar el procesador de *hardware* para suspender el intento;

en respuesta a la suspensión del intento, utilizar el procesador de *hardware* para buscar en la cola de solicitudes una segunda solicitud de análisis formulada de acuerdo con la entidad objetivo, indicando la segunda solicitud de análisis una tercera condición;

en respuesta a la búsqueda de la segunda solicitud de análisis, determinar si se cumple la tercera condición de acuerdo con la entidad objetivo;

en respuesta, cuando se cumple la tercera condición, utilizar el procesador de *hardware* para terminar la entidad objetivo.

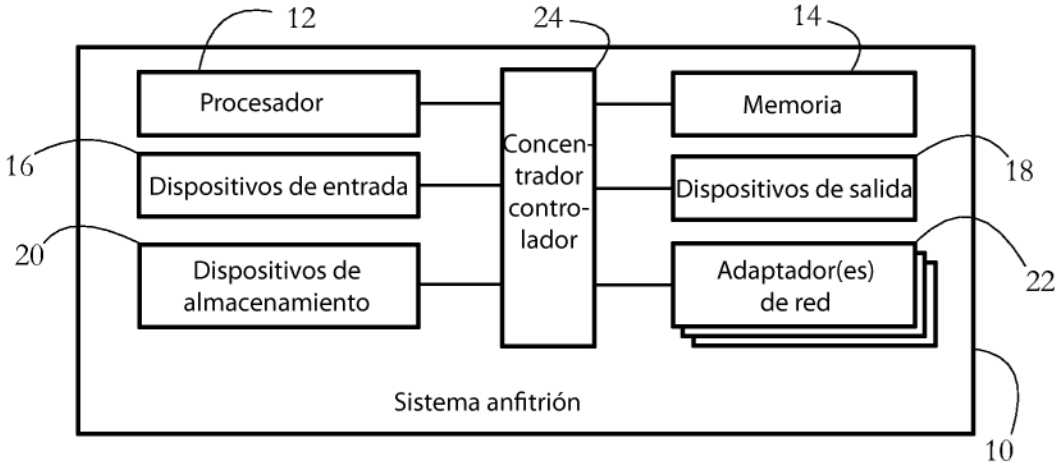
12
24
14
Procesador
Memoria
16
Dispositivos de entrada
Concen-
trador
contro-
lador
Dispositivos de salida
18
20
Dispositivos de
almacenamiento
Adaptador(es)
de red
22
Sistema anfitrión
10

FIG. 1

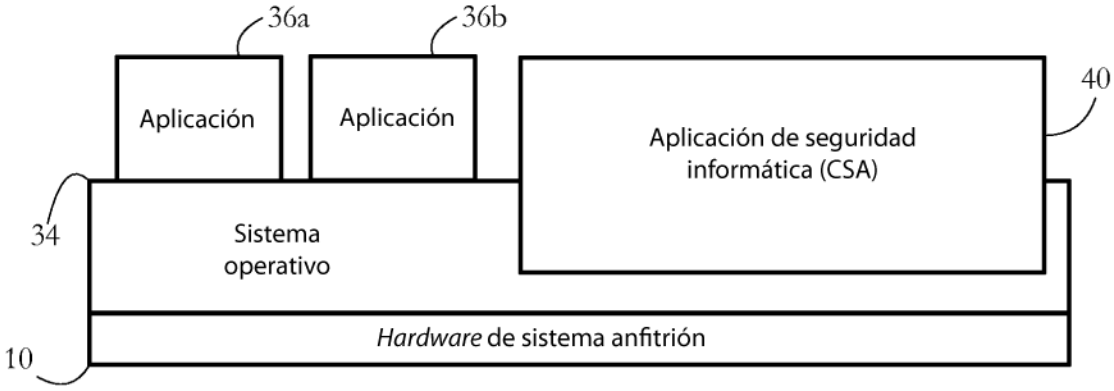
36a
36b
Aplicación
Aplicación
40
Aplicación de seguridad
informática (CSA)
34
Sistema
operativo
Hardware de sistema anfitrión
10

FIG. 2-A

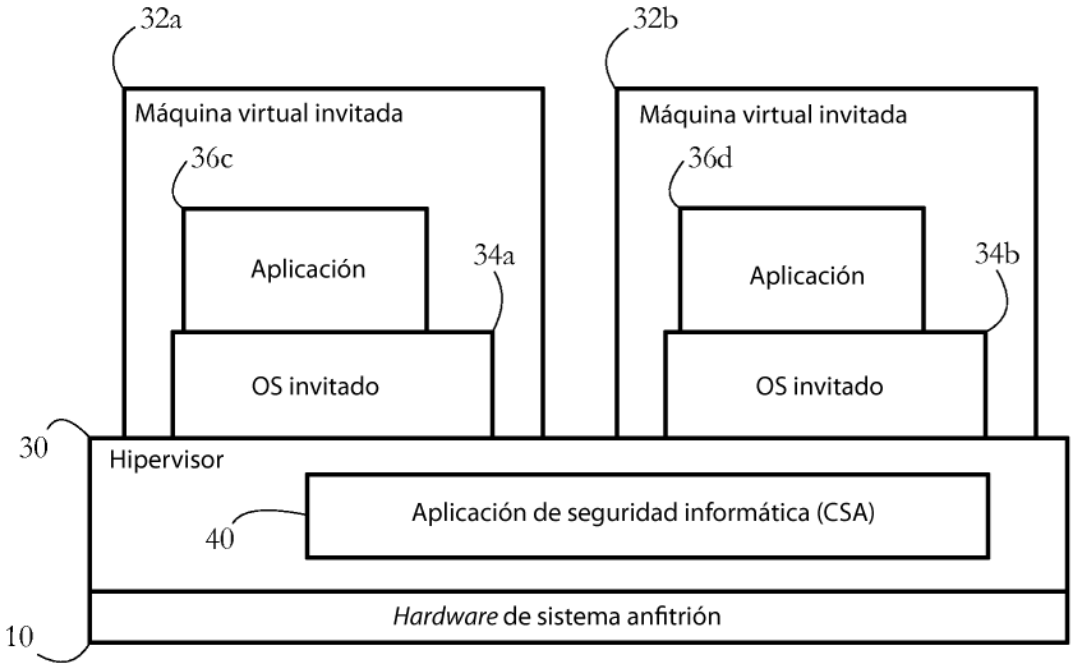
32a
32b
Máquina virtual invitada
Máquina virtual invitada
36c
36d
Aplicación
Aplicación
34a
34b
OS invitado
OS invitado
30
Hipervisor
40
Aplicación de seguridad informática (CSA)
Hardware de sistema anfitrión
10

FIG. 2-B

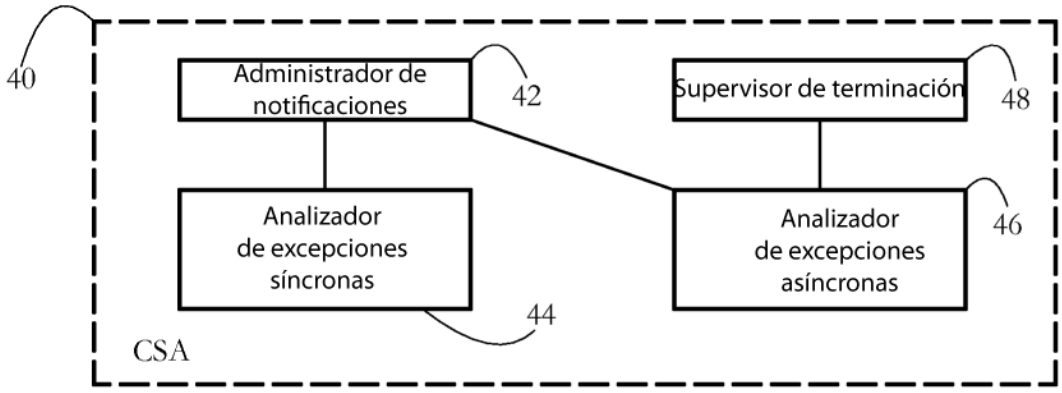
40
Administrador de notificaciones
42
Supervisor de terminación
48
Analizador de excepciones síncronas
Analizador de excepciones asíncronas
46
44
CSA

FIG. 3

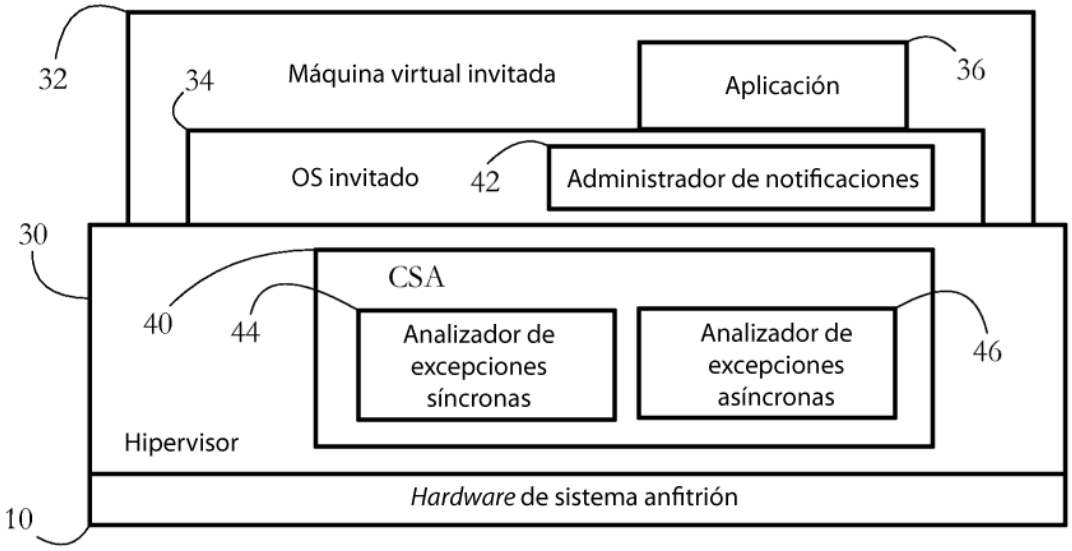
32
34
Máquina virtual invitada
Aplicación
36
OS invitado
42
Administrador de notificaciones
30
CSA
40
44
Analizador de excepciones síncronas
Analizador de excepciones asíncronas
46
Hipervisor
Hardware de sistema anfitrión
10

FIG. 4-A

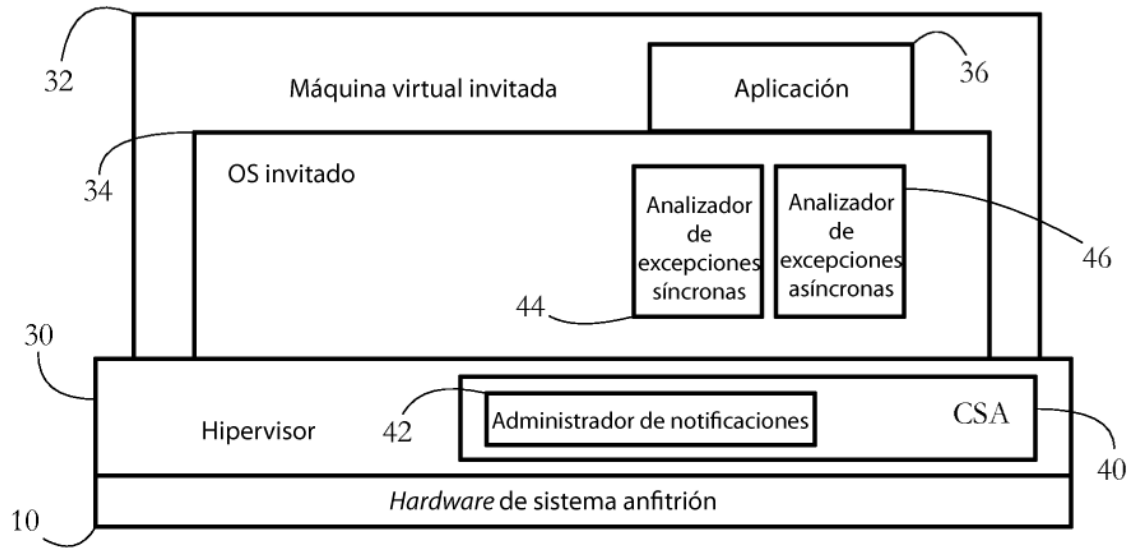
32
Máquina virtual invitada
Aplicación
36
34
OS invitado
Analizador de excepciones síncronas
Analizador de excepciones asíncronas
46
44
30
Hipervisor
42
Administrador de notificaciones
CSA
40
Hardware de sistema anfitrión
10

FIG. 4-B

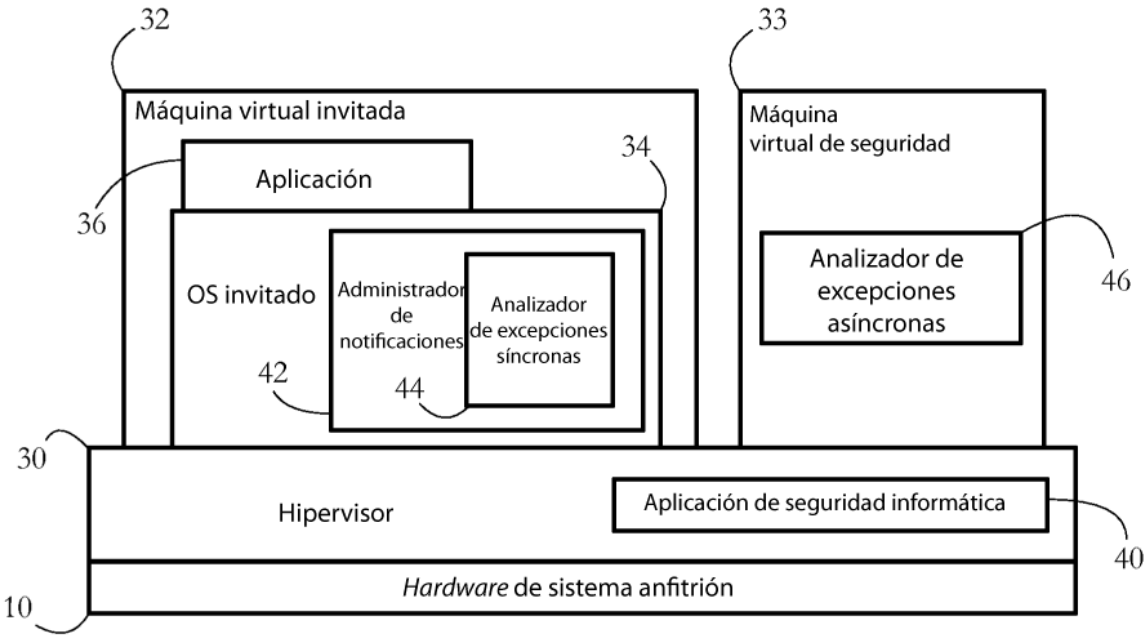
32
33
Máquina virtual invitada
Máquina
virtual de seguridad
34
Aplicación
36
OS invitado
Administrador
de
notificaciones
Analizador
de excepciones
síncronas
Analizador de
excepciones
asíncronas
46
42
44
30
Hipervisor
Aplicación de seguridad informática
40
Hardware de sistema anfitrión
10

FIG. 4-C

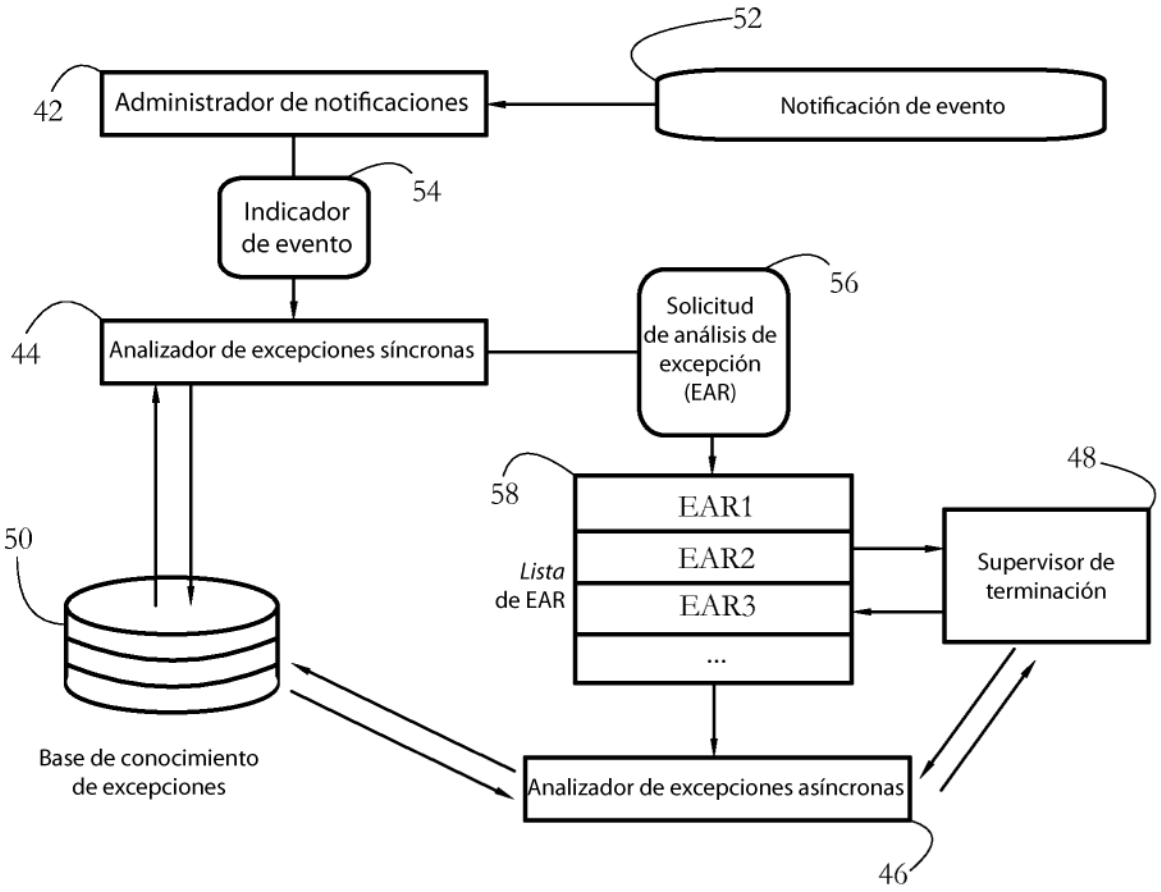
52
42
Administrador de notificaciones
Notificación de evento
54
Indicador de evento
56
44
Analizador de excepciones síncronas
Solicitud de análisis de excepción (EAR)
48
58
EAR1
EAR2
EAR3
...
Lista de EAR
Supervisor de terminación
50
Base de conocimiento de excepciones
Analizador de excepciones asíncronas
46

FIG. 5

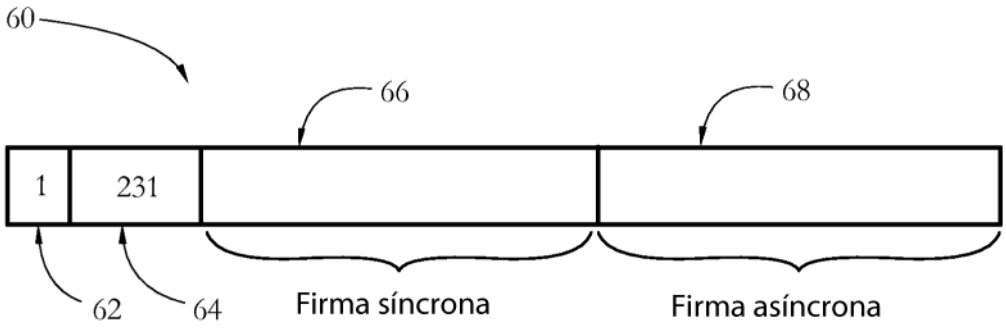
60
66
68
1
231
62
64
Firma síncrona
Firma asíncrona

FIG. 6

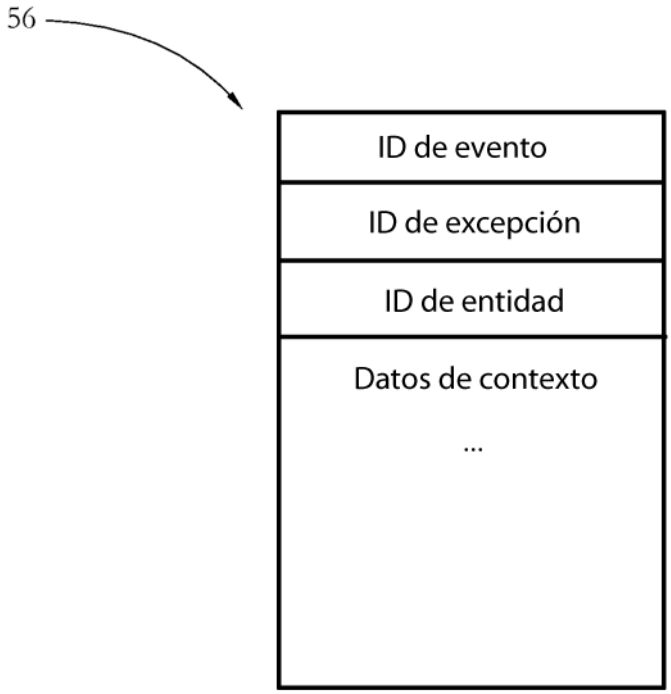
56
ID de evento
ID de excepción
ID de entidad
Datos de contexto
...

FIG. 7

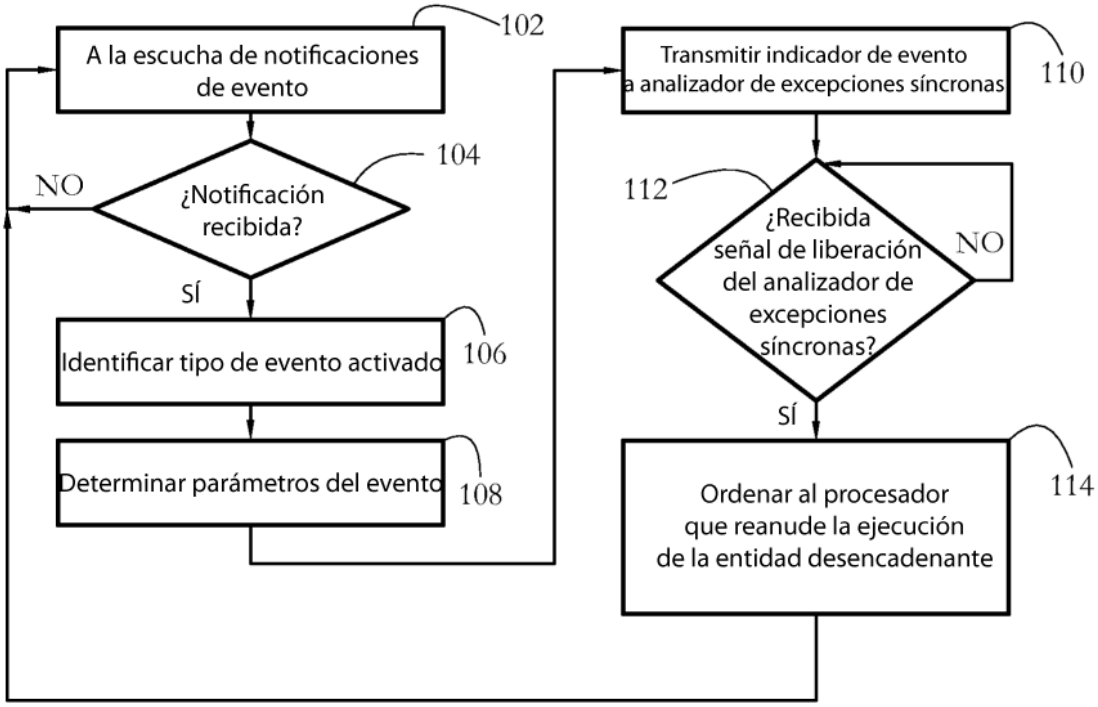
A la escucha de notificaciones de evento
102
¿Notificación recibida?
104
NO
SÍ
Identificar tipo de evento activado
106
Determinar parámetros del evento
108
Transmitir indicador de evento a analizador de excepciones síncronas
110
¿Recibida señal de liberación del analizador de excepciones síncronas?
112
NO
SÍ
Ordenar al procesador que reanude la ejecución de la entidad desencadenante
114

FIG. 8

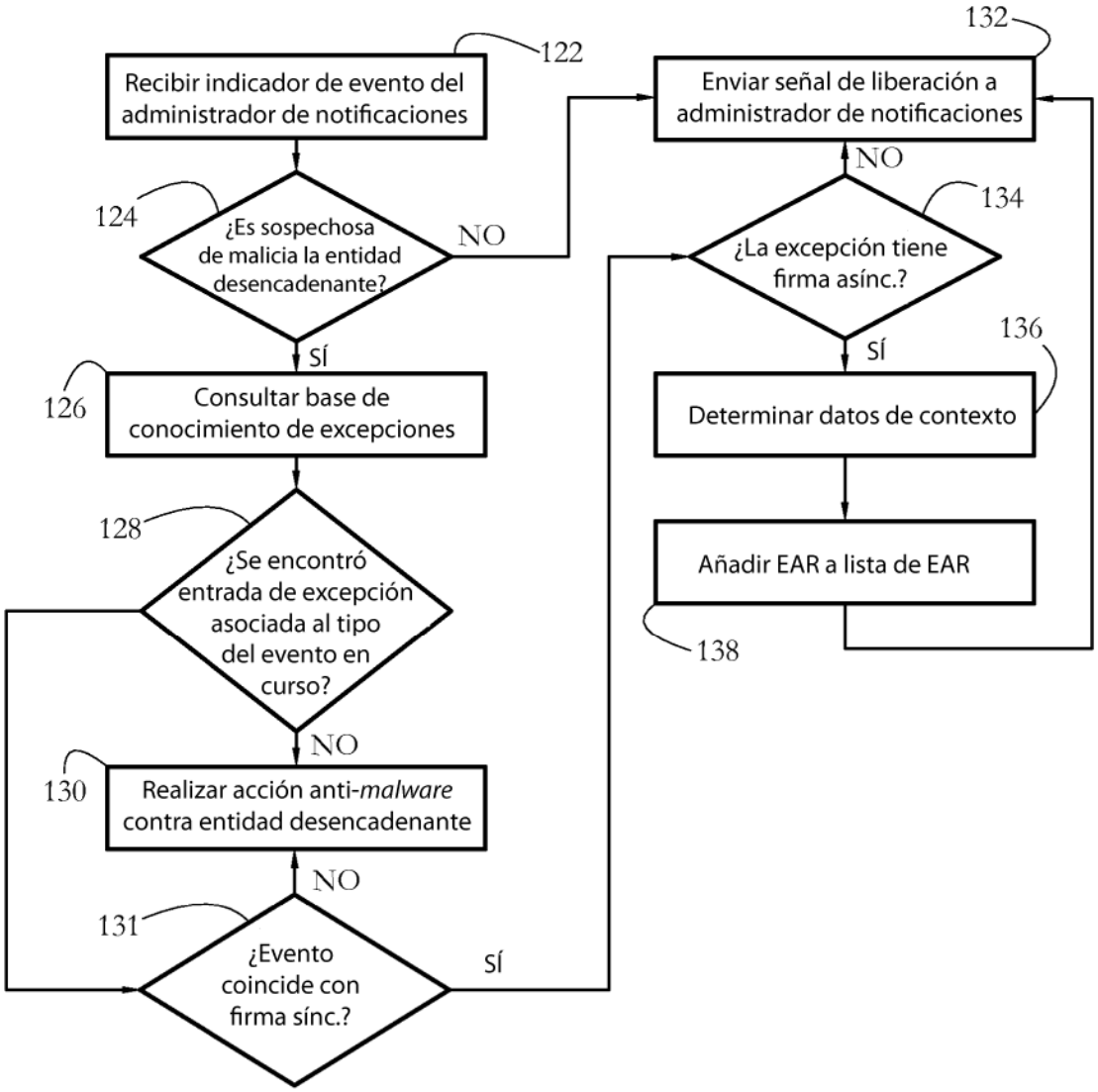
Recibir indicador de evento del administrador de notificaciones
122
¿Es sospechosa de malicia la entidad desencadenante?
124
NO
SÍ
Consultar base de conocimiento de excepciones
126
¿Se encontró entrada de excepción asociada al tipo del evento en curso?
128
NO
Realizar acción anti-malware contra entidad desencadenante
130
NO
¿Evento coincide con firma sínc.?
131
SÍ
Enviar señal de liberación a administrador de notificaciones
132
NO
¿La excepción tiene firma asínc.?
134
SÍ
Determinar datos de contexto
136
Añadir EAR a lista de EAR
138

FIG. 9

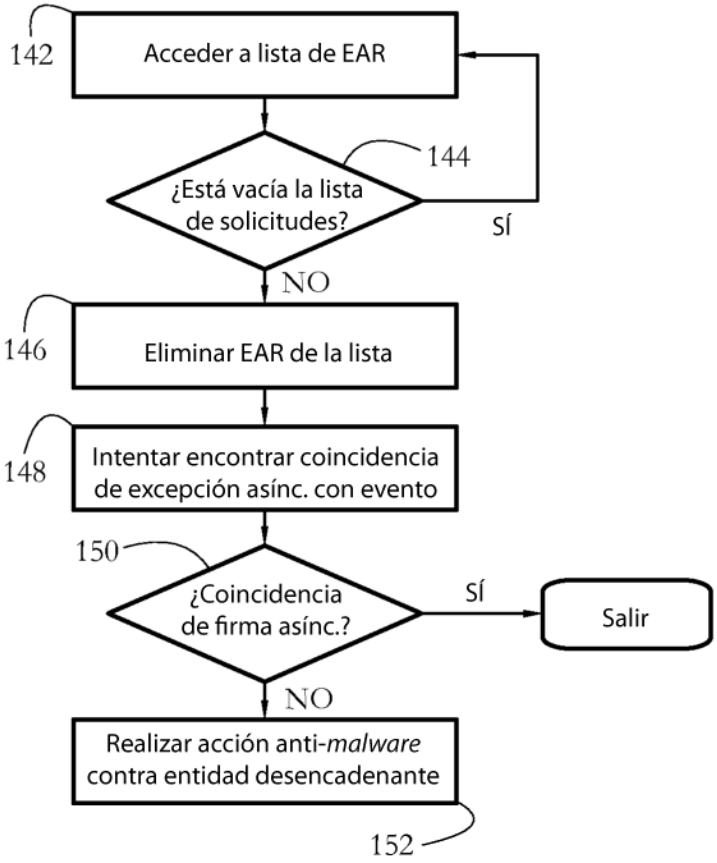
142
Acceder a lista de EAR
144
¿Está vacía la lista de solicitudes?
SÍ
NO
146
Eliminar EAR de la lista
148
Intentar encontrar coincidencia de excepción asínc. con evento
150
¿Coincidencia de firma asínc.?
SÍ
Salir
NO
Realizar acción anti-*malware* contra entidad desencadenante
152

FIG. 10

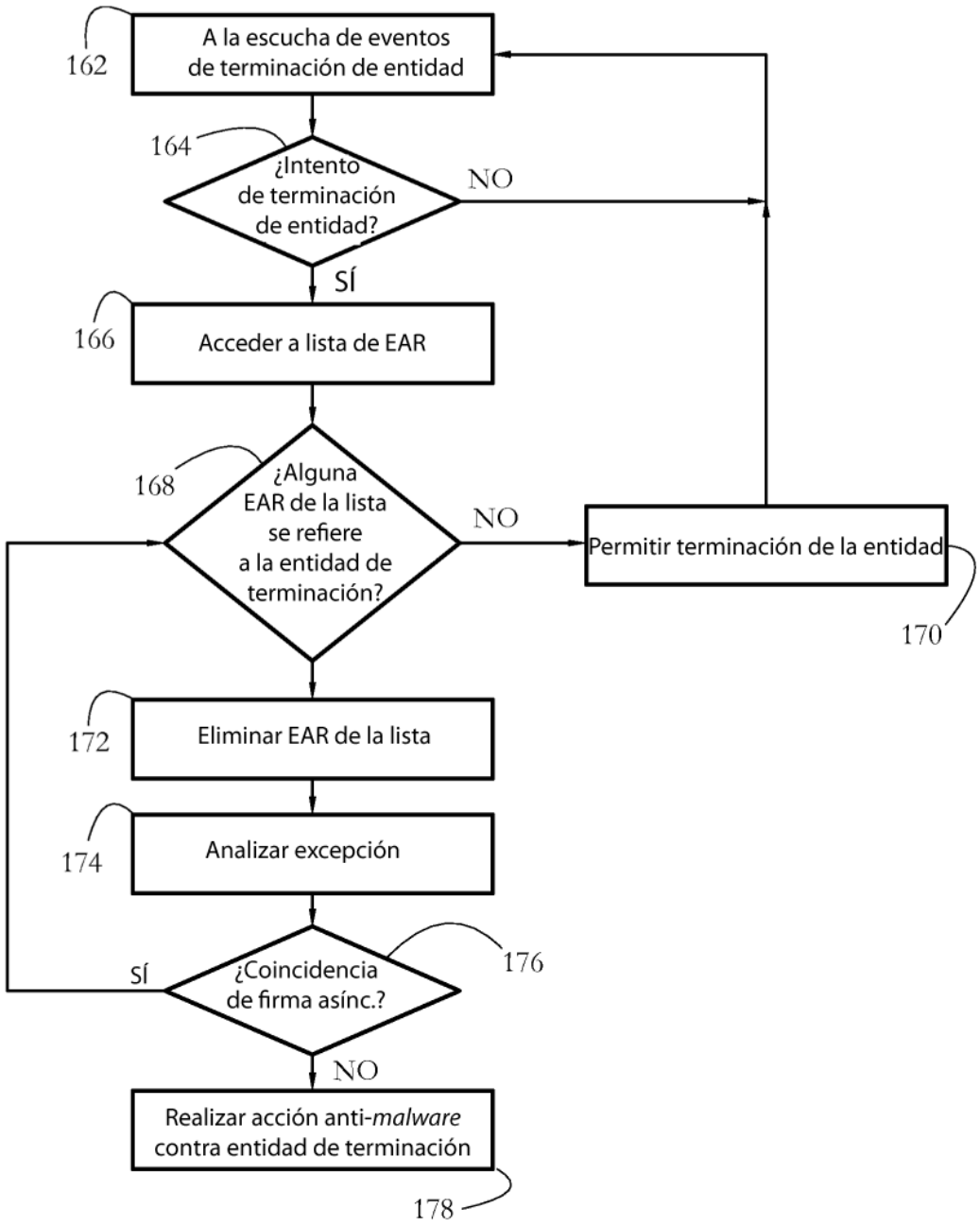
162
A la escucha de eventos de terminación de entidad
164
¿Intento de terminación de entidad?
NO
SÍ
166
Acceder a lista de EAR
168
¿Alguna EAR de la lista se refiere a la entidad de terminación?
NO
Permitir terminación de la entidad
170
172
Eliminar EAR de la lista
174
Analizar excepción
176
¿Coincidencia de firma asínc.?
SÍ
NO
Realizar acción anti-*malware* contra entidad de terminación
178

FIG. 11