(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 785 350**

(51) Int. CI.:

G06F 21/53 (2013.01)
G06F 21/56 (2013.01)
G06F 9/455 (2008.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **02.07.2014 PCT/RO2014/000019**

(87) Fecha y número de publicación internacional: **08.10.2015 WO15152748**

(96) Fecha de presentación y número de la solicitud europea: **02.07.2014 E 14882802 (3)**

(97) Fecha y número de publicación de la concesión europea: **22.01.2020 EP 3017392**

(54) Título: **Evaluación de procesos para la detección de programas malignos en máquinas virtuales**

(30) Prioridad:

05.07.2013 US 201313936058

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
06.10.2020

(73) Titular/es:

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

(72) Inventor/es:

LUKACS, SANDOR;
TOSA, RAUL-VASILE;
BOCA, PAUL-DANIEL;
HAJMASAN, GHEORGHE-FLORIN y
LUTAS, ANDREI-VLAD

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Evaluación de procesos para la detección de programas malignos en máquinas virtuales

Antecedentes

La invención se refiere a sistemas y métodos para proteger sistemas informáticos contra programas malignos y, en particular, a sistemas anti-programas malignos que emplean tecnología de virtualización de hardware.

El software malicioso, también conocido como programa maligno, afecta a una gran cantidad de sistemas informáticos en todo el mundo. En sus muchas formas, tales como virus informáticos, gusanos y herramientas de sustitución de ficheros del sistema raíz, un programa maligno presenta un grave riesgo para millones de usuarios informáticos, haciéndoles vulnerables a la pérdida de datos y de información sensible, al robo de identidad y a la pérdida de productividad, entre otros.

La tecnología de virtualización de hardware permite la creación de entornos informáticos simulados comúnmente conocidos como máquinas virtuales, que se comportan en muchas formas como sistemas informáticos físicos. En aplicaciones típicas tales como la consolidación de servidores y la infraestructura como servicio (IAAS), varias máquinas virtuales se pueden ejecutar simultáneamente en la misma máquina física, compartiendo los recursos de hardware entre ellas, reduciendo de este modo los costes de inversión y operación. Cada máquina virtual puede ejecutar su propio sistema operativo y/o aplicaciones de software, de manera separada de otras máquinas virtuales. Debido a la constante proliferación de programas malignos, cada máquina virtual que opera en tal entorno requiere potencialmente protección contra programas malignos.

Una solución de virtualización comúnmente usada en la técnica comprende un hipervisor, también conocido como un monitor de máquina virtual, que consiste en una capa de software que opera entre el hardware informático y el sistema operativo (OS) de una máquina virtual, y que tiene más privilegios de procesador que el OS respectivo. Dado que algunos programas malignos, tales como las herramientas de sustitución de ficheros del sistema raíz, operan en el nivel de privilegio del OS, hay interés en desarrollar soluciones anti-programas malignos que se ejecuten en el nivel de privilegio del hipervisor.

Un sistema ejemplar diseñado para proteger una plataforma de virtualización contra programas malignos se describe en la solicitud de patente de EE.UU. con número de publicación 2012/0254993 A1, creada por A. S. Sallam, titulada "System And Methods For Virtual Machine Monitor Based Anti Malware Security". El sistema descrito en la misma comprende un agente de seguridad configurado para ejecutarse a un nivel por debajo de todos los sistemas operativos, y un monitor de máquina virtual configurado para interceptar una solicitud de un recurso hecha desde un nivel por encima del monitor de máquina virtual e informar al agente de seguridad acerca de la solicitud. El agente de seguridad está configurado además para determinar si la solicitud es indicativa de un programa maligno.

Compendio

Según un aspecto, un sistema central comprende al menos un procesador configurado para ejecutar: un hipervisor configurado para exponer una máquina virtual; un evaluador de proceso que se ejecuta dentro de la máquina virtual; un motor de introspección de memoria que se ejecuta fuera de la máquina virtual; y un módulo de puntuación de proceso. El evaluador de proceso está configurado para determinar si un proceso evaluado que se ejecuta dentro de la máquina virtual realiza una acción y, en respuesta, cuando el proceso evaluado realiza la acción, transmitir un primer indicador de evaluación de proceso al módulo de puntuación de proceso, el primer indicador de evaluación de proceso determinado para el proceso evaluado. El motor de introspección de memoria está configurado para interceptar una llamada a una función del sistema operativo, para detectar el lanzamiento de un proceso protegido que se ejecuta dentro de la máquina virtual, en donde la función del sistema operativo está configurada para añadir el proceso protegido a una lista de procesos que se ejecutan dentro de la máquina virtual, y en respuesta a la detección del lanzamiento, determinar si el proceso evaluado intenta modificar una página de memoria del proceso protegido y, en respuesta, cuando el proceso evaluado intenta modificar la página de memoria, transmitir un segundo indicador de evaluación de proceso al módulo de puntuación de proceso, el segundo indicador de evaluación de proceso determinado para el proceso evaluado. El módulo de puntuación de proceso está configurado para recibir el primer y segundo indicadores de evaluación de proceso y, en respuesta, determinar si el proceso evaluado es malicioso según el primer y segundo indicadores de evaluación de proceso.

Según otro aspecto, un medio legible por ordenador no transitorio codifica instrucciones que, cuando se ejecutan en un sistema central que comprende al menos un procesador, hacen que el sistema central forme: un hipervisor configurado para exponer una máquina virtual; un evaluador de proceso que se ejecuta dentro de la máquina virtual; un motor de introspección de memoria que se ejecuta fuera de la máquina virtual; y un módulo de puntuación de proceso. El evaluador de proceso está configurado para determinar si un proceso evaluado que se ejecuta dentro de la máquina virtual realiza una acción y, en respuesta, cuando el proceso evaluado realiza la acción, transmitir un primer indicador de evaluación de proceso al módulo de puntuación de proceso, el primer indicador de evaluación de proceso determinado para el proceso evaluado. El motor de introspección de memoria está configurado para interceptar una llamada a una función del sistema operativo, detectar el lanzamiento de un proceso protegido que se ejecuta dentro de la máquina virtual, en donde la función del sistema operativo está configurada para añadir el

proceso protegido a una lista de procesos que se ejecutan dentro de la máquina virtual y, en respuesta a la detección del lanzamiento, determinar si el proceso evaluado intenta modificar una página de memoria del proceso protegido y, en respuesta, cuando el proceso evaluado intenta modificar la página de memoria, transmitir un segundo indicador de evaluación de proceso al módulo de puntuación de proceso, el segundo indicador de evaluación de proceso determinado para el proceso evaluado. El módulo de puntuación de proceso está configurado para recibir el primer y segundo indicadores de evaluación de proceso y, en respuesta, determinar si el proceso evaluado es malicioso según el primer y segundo indicadores de evaluación de proceso.

Según otro aspecto, un método comprende emplear al menos un procesador de un sistema central para recibir un primer indicador de evaluación de proceso determinado para un proceso evaluado, el proceso evaluado que se ejecuta dentro de una máquina virtual expuesta por un hipervisor que se ejecuta en el sistema central. El método comprende además emplear el al menos un procesador para recibir un segundo indicador de evaluación de proceso determinado para el proceso evaluado, y en respuesta a recibir el primer y segundo indicadores de evaluación de proceso, emplear al menos un procesador para determinar si el proceso evaluado es malicioso según el primer y segundo indicadores de evaluación de proceso. La determinación del primer indicador de evaluación de proceso comprende emplear un evaluador de proceso que se ejecuta dentro de la máquina virtual para determinar si el proceso evaluado realiza una primera acción. La determinación del segundo indicador de evaluación de proceso comprende emplear un motor de introspección de memoria que se ejecuta fuera de la máquina virtual para determinar si el proceso evaluado realiza una segunda acción.

Según otro aspecto, un método comprende emplear al menos un procesador de un sistema central para ejecutar un motor de introspección de memoria, el motor de introspección de memoria se ejecuta fuera de una máquina virtual expuesta por un hipervisor que se ejecuta en el sistema central, en donde la ejecución del motor de introspección de memoria comprende la detección del lanzamiento de un proceso que se ejecuta dentro de la máquina virtual. El método comprende además, en respuesta al motor de introspección de memoria que detecta el lanzamiento del proceso, emplear el al menos un procesador para determinar un primer y un segundo indicadores de evaluación de proceso. El método comprende además, en respuesta a la determinación del primer y segundo indicadores de evaluación, emplear el al menos un procesador para determinar si el proceso es malicioso según el primer y segundo indicadores de evaluación de proceso.

Breve descripción de los dibujos

Los aspectos y ventajas anteriores de la presente invención llegarán a ser entendidos mejor tras leer la siguiente descripción detallada y tras hacer referencia a los dibujos donde:

La Figura 1 muestra una configuración de hardware ejemplar de un sistema informático central protegido contra programas malignos según algunas realizaciones de la presente invención.

La Figura 2 muestra un conjunto ejemplar de máquinas virtuales expuestas por un hipervisor que se ejecuta en el sistema central de la Figura 1, y una aplicación de seguridad que opera junto con un motor de introspección de memoria para proteger una máquina virtual según algunas realizaciones de la presente invención.

La Figura 3 ilustra una jerarquía ejemplar de objetos de software que se ejecutan en el sistema central en diversos niveles de privilegio de procesador, que incluyen un conjunto de objetos anti-programas malignos según algunas realizaciones de la presente invención.

La Figura 4 muestra un módulo de puntuación de proceso ejemplar que recibe una pluralidad de indicadores de evaluación de proceso determinados para un proceso por una pluralidad de evaluadores de proceso, según algunas realizaciones de la presente invención.

La Figura 5 muestra una secuencia ejemplar de pasos realizados por el módulo de puntuación de proceso de la Figura 4 según algunas realizaciones de la presente invención.

La Figura 6 muestra una correlación ejemplar de direcciones de memoria en la configuración del sistema de la Figura 2, según algunas realizaciones de la presente invención.

La Figura 7 ilustra un flujo de ejecución ejemplar de un conjunto de procesos en un entorno Windows®. Las flechas continuas indican un flujo de ejecución ejemplar en ausencia de un sistema anti-programas malignos. Las flechas discontinuas indican modificaciones al flujo de ejecución, las modificaciones introducidas por una pluralidad de evaluadores de proceso que operan según algunas realizaciones de la presente invención.

La Figura 8 ilustra una secuencia ejemplar de pasos realizados por el motor de introspección de memoria de las Figuras 2-3 según algunas realizaciones de la presente invención.

La Figura 9 muestra una secuencia ejemplar de pasos realizados por el motor de introspección de memoria para proteger una página de memoria según algunas realizaciones de la presente invención.

La Figura 10 ilustra una configuración ejemplar que comprende una pluralidad de sistemas centrales conectados a un servidor de seguridad a través de una red informática.

La Figura 11 muestra una transacción anti-programas malignos ejemplar entre un sistema central y un servidor de seguridad según algunas realizaciones de la presente invención.

Descripción detallada de realizaciones preferidas

En la siguiente descripción, se entiende que todas las conexiones relatadas 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 relato de un elemento se refiere al menos a un elemento. Una pluralidad de elementos incluye al menos dos elementos. A menos que se requiera de otro modo, los pasos de cualquier método descrito no necesitan ser realizados 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 procesando el segundo elemento y opcionalmente otros datos. Tomar una determinación o decisión según un parámetro abarca tomar la determinación o decisión según el parámetro y opcionalmente según otros datos. A menos que se especifique de otro modo, un indicador de alguna cantidad/datos puede ser la cantidad/datos en sí misma, o un indicador diferente de la cantidad/datos en sí misma. A menos que se especifique de otro modo, un proceso representa una instancia de un programa informático, en donde un programa informático es una secuencia de instrucciones que determinan que un sistema informático realice una tarea específica. A menos que se especifique de otro modo, una página representa la unidad más pequeña de memoria física virtualizada correlacionada individualmente con una memoria física de un sistema informático. Los medios legibles por ordenador abarcan medios no transitorios, tales como medios de almacenamiento magnéticos, ópticos y de semiconductores (por ejemplo, discos duros, discos ópticos, memoria rápida, DRAM), así como enlaces de comunicaciones tales como cables conductores y enlaces de fibra óptica. Según algunas realizaciones, la presente invención proporciona, entre otras cosas, sistemas informáticos que comprenden hardware (por ejemplo, uno o más procesadores) programado para realizar los métodos descritos en la presente memoria, así como instrucciones de codificación de medios legibles por ordenador para realizar los métodos descritos en la presente memoria.

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

La Figura 1 muestra una configuración de hardware ejemplar de un sistema 10 central que realiza operaciones anti-programas malignos según algunas realizaciones de la presente invención. El sistema 10 central puede representar un dispositivo informático corporativo, tal como un servidor empresarial, o un dispositivo de usuario final, tal como un ordenador personal o un teléfono inteligente, entre otros. Otros sistemas centrales incluyen dispositivos de entretenimiento tales como televisores y consolas de juegos, o cualquier otro dispositivo que tenga memoria y un procesador que soporte la virtualización, y que requiera protección contra programas malignos. La Figura 1 muestra un sistema informático con propósitos ilustrativos; otros dispositivos de cliente, tales como teléfonos móviles o tabletas, pueden tener una configuración diferente. En algunas realizaciones, el sistema 10 comprende un conjunto de dispositivos físicos, que incluyen un procesador 12, una unidad 14 de memoria, un conjunto 16 de dispositivos de entrada, un conjunto 18 de dispositivos de salida, un conjunto 20 de dispositivos de almacenamiento y un conjunto 22 de adaptadores de red, todos conectados por un conjunto 24 de buses.

En algunas realizaciones, el procesador 12 comprende un dispositivo físico (por ejemplo, un circuito integrado de múltiples núcleos) configurado para ejecutar operaciones de cálculo y/o lógicas con un conjunto de señales y/o datos. En algunas realizaciones, tales operaciones lógicas se entregan al procesador 12 en forma de una secuencia de instrucciones de procesador (por ejemplo, código máquina u otro tipo de software). La unidad 14 de memoria puede comprender medios legibles por ordenador volátiles (por ejemplo, RAM) que almacenan datos/señales accedidos o generados por el procesador 12 en el curso de llevar a cabo las instrucciones. Los dispositivos 16 de entrada pueden incluir teclados de ordenador, ratones y micrófonos, entre otros, incluyendo las interfaces y/o adaptadores de hardware respectivos que permiten a un usuario introducir datos y/o instrucciones en el sistema 10. Los dispositivos 18 de salida pueden incluir dispositivos de visualización como monitores y altavoces, entre otros, así como interfaces/adaptadores de hardware, tales como tarjetas gráficas, que permiten al sistema 10 comunicar datos a un usuario. En algunas realizaciones, los dispositivos 16 de entrada y los dispositivos 18 de salida pueden compartir una pieza de hardware común, como en el caso de los dispositivos de pantalla táctil. Los dispositivos 20 de almacenamiento incluyen medios legibles por ordenador que permiten el almacenamiento no volátil, la lectura y la escritura de instrucciones software y/o datos. Los dispositivos 20 de almacenamiento ejemplares incluyen discos magnéticos y ópticos y dispositivos de memoria rápida, así como medios extraíbles tales como discos y unidades de CD y/o DVD. El conjunto de adaptadores 22 de red permite que el sistema 10 se conecte con una red informática y/o con otros dispositivos/sistemas informáticos. Los buses 24 representan colectivamente la pluralidad de buses de sistema, de periféricos y de conjunto de chips, y/o toda la demás circuitería que permite la intercomunicación de dispositivos 12-22 del sistema 10 central. Por ejemplo, los buses 24 pueden comprender el puente norte que conecta el procesador 12 con la memoria 14, y/o el puente sur que conecta el procesador 12 con los dispositivos 16-22, entre otros.

La Figura 2 muestra un conjunto ejemplar de máquinas virtuales 32a-b invitadas que se ejecutan en el sistema 10 central y que son expuestas por un hipervisor 30 según algunas realizaciones de la presente invención. Las máquinas virtuales (VM) se conocen comúnmente en la técnica como emulaciones software de máquinas físicas/sistemas informáticos reales, cada una capaz de ejecutar su propio sistema operativo y software independientemente de otras VM. El hipervisor 30 comprende software que permite la multiplexación (compartición) por múltiples máquinas virtuales de recursos de hardware del sistema 10 central, tales como operaciones de procesador, memoria, almacenamiento, entrada/salida y dispositivos de red. En algunas realizaciones, el hipervisor 30 permite que múltiples máquinas virtuales y/o sistemas operativos (OS) se ejecuten concurrentemente en el sistema 10 central, con diversos grados de aislamiento. Para habilitar tales configuraciones, el software que forma parte del hipervisor 30 puede crear una pluralidad de dispositivos virtualizados, es decir, emulados por software, cada dispositivo virtualizado que emula un dispositivo de hardware físico del sistema 10, tal como el procesador 12 y la memoria 14, entre otros. El hipervisor 30 puede asignar además un conjunto de dispositivos virtuales a cada VM que opera en el sistema 10 central. De este modo, cada VM 32a-b opera como si poseyera su propio conjunto de dispositivos físicos, es decir, como un sistema informático más o menos completo. Ejemplos de hipervisores populares incluyen VMware vSphere™ de VMware Inc. y el hipervisor de código abierto Xen, entre otros.

En algunas realizaciones, el hipervisor 30 incluye un motor 40 de introspección de memoria, configurado para realizar operaciones anti-programa maligno como se describe más adelante. El motor 40 se puede incorporar en el hipervisor 30, o se puede entregar como un componente software distinto e independiente del hipervisor 30, pero que se ejecuta a un nivel de privilegio de procesador sustancialmente similar al hipervisor 30. Un único motor 40 se puede configurar para proteger contra programas malignos múltiples VM que se ejecutan en el sistema 10 central.

Aunque la Figura 2 muestra sólo dos máquinas virtuales 32a-b por simplicidad, el sistema 10 central puede operar un gran número, por ejemplo, cientos, de VM concurrentemente, y el número de tales VM puede cambiar durante la operación del sistema 10 central. En algunas realizaciones, cada VM 32a-b ejecuta un sistema operativo 34a-b invitado y/o un conjunto de aplicaciones 42a-b, 42c y 44 software, respectivamente, concurrente e independientemente de otras VM que se ejecutan en el sistema 10 central. Cada OS 34a-b comprende software que proporciona una interfaz para el hardware (virtualizado) de la VM 32a-b respectiva, y actúa como un ordenador central para aplicaciones software que se ejecutan en el OS respectivo. Los sistemas operativos 34a-b pueden comprender cualquier sistema operativo ampliamente disponible tal como Windows®, MacOS®, Linux®, iOS® o Android™, entre otros. Las aplicaciones 42a-c pueden incluir procesadores de texto, procesadores de imágenes, base de datos, navegadores y aplicaciones de comunicación electrónica, entre otras. En la siguiente descripción, se dice que el software que se ejecuta en un procesador virtual de una máquina virtual se ejecuta dentro de la máquina virtual respectiva. Por ejemplo, en la Figura 2, se dice que la aplicación 42b se ejecuta dentro de la VM 32a, mientras que se dice que la aplicación 42c se ejecuta dentro de la VM 32b. Por el contrario, se dice que el motor 40 de introspección de memoria se ejecuta fuera de las VM 32a-b.

En el ejemplo de la Figura 2, una aplicación 44 de seguridad se ejecuta en el OS 34b invitado, la aplicación 44 configurada para realizar operaciones anti-programas malignos (AM) junto con el motor 40 de introspección de memoria, como se detalla a continuación, para proteger la máquina virtual 32b del programa maligno. En algunas realizaciones, una instancia de la aplicación 44 se puede ejecutar en cada una de una pluralidad de VM que operan en el sistema 10 central, cada una de tales instancias configurada para interactuar con el motor 40 de introspección para proteger la máquina virtual respectiva. La aplicación 44 de seguridad puede ser un programa autónomo o puede formar parte de un paquete de software que comprende, entre otros, componentes anti-programa maligno, anti-correo electrónico no deseado y anti-software espía.

La Figura 3 ilustra una jerarquía de objetos de software que se ejecutan en un sistema 10 central según algunas realizaciones de la presente invención. La Figura 3 está representada desde la perspectiva de los niveles de privilegio del procesador, también conocidos en la técnica como capas o anillos de protección. En algunas realizaciones, cada capa o anillo de protección tal se caracteriza por un conjunto de instrucciones, que un objeto de software que se ejecuta en el nivel de privilegio de procesador respectivo, se permite que se ejecute. Cuando un objeto de software intenta ejecutar una instrucción que no está permitida dentro del nivel de privilegio respectivo, el intento puede desencadenar un evento de procesador, tal como una excepción, un fallo o un evento de salida de máquina virtual. En algunas realizaciones, la conmutación entre niveles de privilegio se puede lograr a través de un conjunto de instrucciones dedicadas. Tales instrucciones ejemplares incluyen SYSCALL/SYSENTER, que conmutan del nivel de usuario al nivel de núcleo, SYSRET/SYSEXIT, que conmutan del nivel de núcleo al nivel de usuario, VMCALL, que conmuta o bien del nivel de usuario o bien de núcleo al nivel de raíz, y VMRESUME, que conmuta del nivel de raíz al nivel de núcleo o de usuario.

En algunas realizaciones, el hipervisor 30 toma el control del procesador 12 en el nivel más privilegiado (por ejemplo, VMXroot en plataformas Intel® que soportan virtualización, y también conocido como anillo -1 o modo raíz), creando de este modo una plataforma de virtualización de hardware presentada como una máquina virtual 32 a otro software que se ejecuta en el sistema 10 central. Un sistema operativo 34, como los OS 34a-b en la Figura 2, se ejecuta dentro del entorno virtual de la VM 32, el OS 34 que tiene un privilegio de procesador menor que el hipervisor 30 (por ejemplo, anillo 0 en plataformas Intel, o modo núcleo). Un conjunto de aplicaciones 42d-e se ejecuta con un privilegio de procesador menor que el OS 34 (por ejemplo, anillo 3 o modo usuario).

En algunas realizaciones, partes de la aplicación 44 de seguridad se pueden ejecutar con privilegios de procesador en el nivel de usuario, es decir, el mismo nivel que las aplicaciones 42d-e. Por ejemplo, tales partes pueden comprender una interfaz gráfica de usuario que informa a un usuario de cualquier programa maligno o de amenazas de seguridad detectadas en la VM respectiva, y que recibe información del usuario que indica, por ejemplo, una opción de configuración deseada para la aplicación 44. Otro ejemplo de un componente que se ejecuta en el nivel de usuario es un evaluador de proceso en el nivel de usuario, como se detalla a continuación. Otras partes de la aplicación 44 se pueden ejecutar en el nivel de privilegio de núcleo. Por ejemplo, la aplicación 44 puede instalar un controlador 36 anti-programa maligno y un módulo 38 de puntuación de proceso, ambos operando en modo núcleo. Un controlador 36 AM ejemplar proporciona funcionalidad a la aplicación 44 anti-programa maligno, por ejemplo, para escanear la memoria para firmas de programa maligno y/o para detectar comportamientos indicativos de programas malignos de procesos y/u otros objetos de software que se ejecutan en el OS 34.

En algunas realizaciones, el módulo 38 de puntuación de proceso está configurado para recibir datos de evaluación de proceso desde una pluralidad de componentes de software, los datos de evaluación de proceso determinados para un proceso evaluado, y para determinar si el proceso evaluado es malicioso según los datos respectivos. Un proceso es una instancia de un programa informático, tal como una aplicación o una parte de un sistema operativo, y se caracteriza por tener al menos un hilo de ejecución y una sección de memoria virtual asignada por el sistema operativo, la sección respectiva que comprende código ejecutable. En algunas realizaciones, el sistema operativo gestiona los procesos que se ejecutan actualmente en el sistema 10 central (o dentro de la máquina virtual 32, en el caso de virtualización), tal gestión que incluye, entre otros, la asignación de memoria virtual a cada proceso y la programación de cada proceso o hilo del mismo para su ejecución.

La Figura 4 muestra un módulo 38 de puntuación de proceso ejemplar que recibe una pluralidad de indicadores 52a-d de evaluación de proceso, cada indicador 52a-d determinado por un componente evaluador de proceso. En la Figura 4, tales componentes de evaluación incluyen un evaluador 50a de proceso en el nivel de usuario, un evaluador 50b de proceso en el nivel de núcleo y un evaluador 50c de llamadas al sistema, entre otros. Los evaluadores 50a-c se pueden poner en marcha por el, o formar parte del, controlador 36 de anti-programas malignos. Cada evaluador tal puede ejecutarse independientemente de otros evaluadores, y cada uno puede determinar una pluralidad de indicadores de evaluación de proceso distintos del proceso evaluado. La operación de los evaluadores 50a-c se detallará más adelante. En algunas realizaciones, algunos indicadores de evaluación de proceso, tales como los indicadores 52a-c en la Figura 4, se determinan por componentes que se ejecutan dentro de la VM 32, mientras que otros indicadores de evaluación de procesos, tales como 52d, se determinan por componentes que se ejecutan fuera de la VM 32 (por ejemplo, por el motor 40 de introspección de memoria).

Algunos indicadores de evaluación pueden ser indicativos de un programa maligno, es decir, pueden indicar que el proceso evaluado es malicioso. Algunos indicadores de evaluación pueden no ser indicativos de un programa maligno por sí mismos, pero pueden indicar malicia cuando se combinan con otros indicadores de evaluación. Cada indicador 52a-d de evaluación se puede determinar según un método o criterio distinto. Un indicador de evaluación de proceso ejemplar determinado para un proceso evaluado puede incluir, por ejemplo, un indicador de comportamiento, que indica si el proceso evaluado realizó o intentó realizar, una cierta acción, tal como editar una clave de registro de sistema de la VM 32 o escribir en una página de memoria que pertenece a un objeto de software protegido. Otro indicador de evaluación de proceso ejemplar puede indicar si una sección de memoria que pertenece al proceso evaluado contiene una firma indicativa de programa maligno. En algunas realizaciones, cada indicador 52a-d de evaluación de proceso comprende un indicador de identificación de proceso, tal como un ID de proceso, una etiqueta, o un índice de comprobación aleatoria, permitiendo que el módulo 38 identifique el proceso para el que se determinó el indicador respectivo.

En algunas realizaciones, un indicador de evaluación de proceso puede comprender una puntuación numérica determinada por el evaluador de proceso respectivo, la puntuación indicativa de un grado de malicia del proceso respectivo. Alternativamente, tales puntuaciones se pueden determinar por el módulo 38 según los indicadores 52a-d de evaluación de proceso. Las puntuaciones de malicia pueden ser binarias (1/0, sí/no) o pueden variar en un intervalo continuo de valores. Una puntuación de malicia ejemplar que puede variar dentro de un intervalo de valores comprende un número indicativo de una verosimilitud (por ejemplo, probabilidad) de que el proceso evaluado sea malicioso; tal puntuación puede variar, por ejemplo, entre 0 y 1, o entre 0% y 100%. Los valores de puntuación pueden ser específicos del comportamiento. Por ejemplo, un proceso evaluado puede recibir una puntuación de malicia de 0.2 cuando crea un archivo de disco, y una puntuación de malicia de 0.7 cuando modifica un valor de registro de Windows.

La Figura 5 muestra una secuencia ejemplar de pasos ejecutados por el módulo 38 de puntuación de proceso según algunas realizaciones de la presente invención. En un paso 302, el módulo 38 recibe un indicador de evaluación de proceso, tal como los indicadores 52a-d en la Figura 4, desde un evaluador de proceso que puede operar o bien dentro de la VM 32 (véanse, por ejemplo, los evaluadores 50a-c en la Figura 4), o bien fuera de la VM 32 (por ejemplo, el motor 40 de introspección de memoria). En un paso 304, el módulo 38 puede identificar el proceso para el cual se determinó el indicador de evaluación de proceso respectivo. En algunas realizaciones, el módulo 38 de puntuación de proceso puede mantener un registro por proceso de todos los indicadores de evaluación de proceso recibidos desde varios evaluadores de proceso; el paso 304 puede comprender además añadir el indicador recibido en el paso 302 al registro del proceso respectivo.

Para determinar si un proceso evaluado es malicioso, en un paso 306, el módulo 38 de puntuación de proceso puede determinar una puntuación agregada combinando puntuaciones individuales determinadas para el proceso respectivo, y recibidas desde varios evaluadores de proceso. Las puntuaciones agregadas ejemplares comprenden una suma ponderada y una media ponderada de puntuaciones individuales. En algunas realizaciones, la puntuación agregada puede combinar indicadores/puntuaciones de evaluación de proceso determinados para el proceso evaluado con indicadores/puntuaciones de evaluación de proceso determinados para otros procesos u objetos de software. Por ejemplo, las puntuaciones determinadas para el proceso evaluado se pueden combinar con las puntuaciones determinadas para un proceso descendiente del proceso evaluado, y/o con las puntuaciones determinadas para un proceso ascendiente del proceso evaluado.

En un paso 308, el módulo 38 puede comparar la puntuación agregada con un umbral predeterminado. Cuando la puntuación agregada no excede el umbral, el módulo 38 puede volver al paso 302 descrito anteriormente. En algunas realizaciones, el umbral se puede establecer en un valor determinado según una entrada recibida desde un usuario de la VM respectiva (por ejemplo, a través de una interfaz de usuario expuesta por la aplicación 44 de seguridad). Los valores de umbral pueden reflejar las preferencias de seguridad del usuario respectivo. Por ejemplo, cuando el usuario opta por una seguridad estricta, el umbral se puede establecer en un valor relativamente bajo; cuando el usuario prefiere una configuración de seguridad más tolerante, el umbral se puede establecer en un valor relativamente alto. En algunas realizaciones, el valor umbral se puede recibir desde un servidor de seguridad remoto, como se describe a continuación en relación con las Figuras 10-11.

En algunas realizaciones, en los pasos 306-308, el módulo 38 de puntuación de proceso puede determinar una pluralidad de puntuaciones agregadas y comparar cada puntuación agregada con un umbral (posiblemente distinto). Cada puntuación agregada tal se puede determinar según un subconjunto distinto de indicadores de evaluación de proceso. En una realización ejemplar, cada conjunto de indicadores de evaluación de proceso tal puede representar una clase o tipo particular de programa maligno (por ejemplo, Troyanos, herramientas de sustitución de ficheros del sistema raíz, etc.), que permiten que el módulo 38 realice una clasificación del programa maligno detectado.

Cuando la puntuación agregada excede el umbral, en un paso 310, el módulo 38 puede decidir que el proceso evaluado es malicioso y puede seguir alguna acción anti-programa maligno. En algunas realizaciones, tal acción anti-programa maligno pueden incluir, entre otras, terminar el proceso evaluado, poner en cuarentena el proceso evaluado y eliminar o deshabilitar un recurso (como un archivo o una sección de memoria) del proceso evaluado. En algunas realizaciones, la acción anti-programa maligno pueden comprender además alertar a un usuario del sistema 10 central y/o alertar a un administrador del sistema, por ejemplo, enviando un mensaje al administrador del sistema sobre una red informática conectada al sistema 10 central a través del adaptador o adaptadores 22 de red. En algunas realizaciones, la acción anti-programa maligno también puede comprender enviar un informe de seguridad a un servidor de seguridad remoto, como se describe a continuación en relación con las Figuras 10-11.

El módulo 38 de puntuación de proceso ejemplar representado en las Figuras 3-4 opera dentro de la VM 32 en el nivel de privilegio de procesador de OS (por ejemplo, modo núcleo). En realizaciones alternativas, el módulo 38 de puntuación de proceso se puede ejecutar dentro de la VM 32 en modo usuario, o incluso fuera de la VM 32, en el nivel de privilegio de procesador del hipervisor 30.

En algunas realizaciones, el motor 40 de introspección se ejecuta sustancialmente en el mismo nivel de privilegio que el hipervisor 30, y está configurado para realizar la introspección de máquinas virtuales tales como la VM 32. La introspección de una VM, o de un objeto de software que se ejecuta en la VM respectiva, puede comprender analizar el comportamiento del objeto de software, determinar y/o acceder a las direcciones de memoria de tales objetos de software, restringir el acceso de ciertos procesos a un contenido de memoria situado en tales direcciones, analizar tal contenido y determinar los indicadores de evaluación de proceso de los respectivos objetos de software (por ejemplo, el indicador 52d en la Figura 4), entre otros. En algunas realizaciones, los objetos de software de destino del motor 40 de introspección comprenden procesos, flujos de instrucciones, registros y estructuras de datos tales como tablas de páginas y objetos de controlador de la VM respectiva, entre otros.

Para realizar la introspección de la VM 32 desde fuera de la VM respectiva, algunas realizaciones del motor 40 emplean estructuras de correlación de memoria y mecanismos del procesador 12. Las máquinas virtuales operan típicamente con una memoria física virtualizada, es decir, una representación virtual de la memoria 14 física real del sistema 10 central. La memoria física virtualizada comprende un espacio contiguo de direcciones virtualizadas, específico para cada VM invitada que se ejecuta en el sistema 10 central, con partes del espacio respectivo correlacionadas con direcciones dentro de la memoria 14 física y/o dispositivos 20 de almacenamiento físico. En los sistemas configurados para soportar la virtualización, tal correlación se logra típicamente mediante estructuras de datos dedicadas controladas por el procesador 12, tales como las tablas de páginas extendidas (EPT) o las tablas de páginas anidadas (NPT). En tales sistemas, la memoria física virtualizada se puede dividir en unidades conocidas en la técnica como páginas. Una página representa la unidad más pequeña de memoria física virtualizada correlacionada individualmente con la memoria física a través de mecanismos tales como la EPT y/o la NPT, es decir, la correlación entre la memoria física y la física virtualizada se realiza con granularidad de página. Todas las páginas tienen típicamente un tamaño predeterminado, por ejemplo, 4 kilobytes, 2 megabytes, etc. La partición de la memoria física virtualizada en páginas se configura normalmente por el hipervisor 30. En algunas realizaciones, el hipervisor 30 también configura la EPT/NPT y, por lo tanto, la correlación entre la memoria física y la memoria física

virtualizada. La traducción real de una dirección de memoria física virtualizada a una dirección de memoria física puede comprender buscar la dirección de memoria física en un almacenador temporal de traducción anticipada (TLB) del sistema 10 central. En algunas realizaciones, la traducción de direcciones comprende realizar un recorrido de página, que incluye un conjunto de búsquedas sucesivas de direcciones en un conjunto de tablas de página, y realizar cálculos tales como añadir un desplazamiento de una página a una dirección con relación a la página respectiva.

Algunas configuraciones de hardware permiten que el hipervisor 30 controle selectivamente el acceso a los datos almacenados dentro de cada página, por ejemplo, estableciendo derechos de acceso de lectura y escritura a la página respectiva. Tales derechos se pueden establecer, por ejemplo, modificando una entrada de la página respectiva dentro de la EPT o de la NPT. El hipervisor 30 puede seleccionar de este modo qué objeto de software puede acceder a los datos almacenados en las direcciones dentro de cada página, y puede indicar qué operaciones están permitidas con los datos respectivos, por ejemplo, leer, escribir, etc. Un intento de un objeto de software que se ejecuta dentro de una VM de realizar una operación, tal como leer datos o escribir datos en una página para la cual el objeto no tiene el derecho respectivo, puede desencadenar un evento de salida de máquina virtual (por ejemplo, un evento VMExit en plataformas Intel). En algunas realizaciones, los eventos de salida de máquina virtual transfieren el control del procesador desde la VM que ejecuta el objeto de software respectivo al hipervisor 30 o al motor 40 de introspección de memoria, permitiendo de esta manera que el hipervisor 30 y/o el motor 40 intercepten y analicen el intento de lectura/escritura no autorizado.

En algunas realizaciones, el OS 34 configura un espacio de memoria virtual (también denominado espacio de direcciones lógicas) y expone el espacio de memoria virtual a una aplicación tal como las aplicaciones 42d-e y 44 en la Figura 3. En tales sistemas, el OS 34 configura y mantiene una correlación entre el espacio de memoria virtual y la memoria física virtualizada de la VM 32, por ejemplo, usando un mecanismo de tabla de páginas. En algunas realizaciones, el espacio de memoria virtual también se divide en páginas, tales páginas que representan la unidad más pequeña de la memoria virtual correlacionada individualmente con la memoria física virtualizada por el OS 34 (la correlación de memoria virtual a la física virtualizada se realiza con granularidad de página).

La Figura 6 ilustra una correlación (traducción) ejemplar de direcciones de memoria en una realización como se muestra en la Figura 2. A un objeto de software, tal como una aplicación o un proceso que se ejecuta dentro de la VM 32a, se le asigna un espacio 214a de dirección virtual por el OS 34a invitado. Cuando el objeto de software respectivo intenta acceder a una dirección 60a de memoria ejemplar del espacio 214a, la dirección 60a se traduce por el procesador virtualizado de la VM 32a invitada, según las tablas de páginas configuradas y controladas por el OS 34a invitado, en una dirección 60b dentro de un espacio 114a de memoria física virtualizada de la máquina virtual 32a. La dirección 60b también se conoce en la técnica como una dirección física invitada. El hipervisor 30, que configura y controla la memoria 114a física virtualizada, correlaciona la dirección 60b con una dirección 60c dentro de la memoria 14 física del sistema 10 central, por ejemplo, usando medios de la EPT o la NPT, como se ha tratado anteriormente.

De manera similar, un espacio 214b de memoria virtual se pone en marcha por el OS 34b invitado para aplicaciones (por ejemplo, 42c) u otros objetos de software que se ejecutan en la VM 32b invitada. Una dirección 60d virtual ejemplar dentro del espacio 214b se traduce por el procesador virtualizado de la VM 32b invitada, según las tablas de páginas configuradas y controladas por el OS 34b invitado, en una dirección 60e dentro de un espacio 114b de memoria física virtualizada de la VM 32b invitada. La dirección 60e se correlaciona además por el hipervisor 30 a una dirección 60f dentro de la memoria 14 física.

En algunas realizaciones, el hipervisor 30 pone en marcha su propio espacio 214c de memoria virtual que comprende una representación de la memoria 14 física, y emplea un mecanismo de traducción (por ejemplo, tablas de páginas) para correlacionar direcciones en el espacio 214c con direcciones en la memoria 14 física. En la Figura 6, tal correlación ejemplar traduce una dirección 60g en una dirección 60h. De manera similar, direcciones tales como 60c y 60f en la memoria 14 física corresponden a las direcciones 60k y 60m, respectivamente, dentro del espacio 214c de memoria virtual del hipervisor 30. Tal traducción permite al hipervisor 30 gestionar (por ejemplo, leer, escribir y controlar el acceso a) las páginas de memoria que pertenecen a objetos de software que se ejecutan dentro de varias máquinas virtuales que se ejecutan en el sistema 10 central.

La Figura 7 ilustra un flujo de ejecución ejemplar de un conjunto de procesos 70a-b que se ejecuta en una VM 32 según algunas realizaciones de la presente invención. El ejemplo de la Figura 7 muestra el flujo de ejecución en un sistema que ejecuta una versión del OS Windows®; se pueden representar diagramas similares para otros sistemas operativos tales como Linux, por ejemplo. Las flechas continuas representan el flujo de ejecución en ausencia de un sistema anti-programas malignos, tal como la aplicación 44 de seguridad. Las flechas discontinuas representan modificaciones en el flujo debido a la presencia de evaluadores de proceso que se ejecutan según algunas realizaciones de la presente invención.

El proceso 70a comprende una pluralidad de bibliotecas de enlace dinámico (DLL) 72a-c; en el ejemplo de la Figura 7, la DLL 72c se inyecta en el proceso 70a por el proceso 70b (posiblemente malicioso). La inyección de código es un término genérico usado en la técnica para indicar una familia de métodos de introducción de una secuencia de código, tal como una DLL, en el espacio de memoria de un proceso existente, para alterar la funcionalidad original

del proceso respectivo. Cuando el proceso 70a ejecuta una instrucción que llama a alguna funcionalidad del sistema, por ejemplo, escribir algo en un archivo de disco o editar una clave de registro, la instrucción respectiva llama a una API de modo de usuario tal como KERNEL32.DLL o NTDLL.DLL. En el ejemplo de la Figura 7, la llamada a la API de modo de usuario respectivo se intercepta y analiza por el filtro 50a de comportamiento a nivel de usuario. Tales intercepciones se pueden lograr mediante un método tal como inyección de DLL o enganche, entre otros. Enganche es un término genérico usado en la técnica para un método de interceptación de llamadas a funciones, o mensajes o eventos pasados entre componentes de software. Un método de enganche ejemplar comprende alterar el punto de entrada de una función objetivo, insertando una instrucción que redirige la ejecución a una segunda función. Después de tal enganche, la segunda función se puede ejecutar en lugar, o antes, que la función objetivo. En el ejemplo de la Figura 7, el controlador 36 anti-programa maligno puede engancharse a ciertas funciones de KERNEL32.DLL o NTDLL.DLL, para instruir a las funciones respectivas para redirigir la ejecución al filtro 50a. De este modo, el filtro 50a puede detectar que el proceso 70a está intentando realizar un cierto comportamiento, identificado según la función que realiza la redirección. Cuando el filtro 50a detecta tal comportamiento, el filtro 50 puede formular el indicador 52a de evaluación del proceso (Figura 4) y transmitir el indicador 52a al módulo 38 de puntuación de proceso.

En un flujo de ejecución típico, la función de API de modo de usuario puede solicitar un servicio del núcleo del sistema operativo. En algunas realizaciones, tales operaciones se realizan emitiendo una llamada al sistema, tal como SYSCALL y SYSENTER en plataformas x86. En el ejemplo de la Figura 7, tales llamadas al sistema se interceptan por el evaluador 50c de llamadas al sistema. En algunas realizaciones, tal interceptación comprende, por ejemplo, modificar una rutina de manejo de llamadas al sistema cambiando un valor almacenado en un registro específico de modelo (MSR) del procesador 12, que redirige efectivamente la ejecución al filtro 50c. Tales técnicas se conocen en la técnica como enganche de MSR, y pueden permitir que el evaluador 50c de llamadas al sistema detecte que el proceso evaluado está intentando realizar ciertas llamadas al sistema. Cuando se interceptan tales llamadas al sistema, el filtro 50c de llamadas al sistema puede formular el indicador 52c de evaluación de proceso y transmitir el indicador 52c al módulo 38 de puntuación de proceso.

Siguiendo a la llamada al sistema, el control del procesador se pasa típicamente al núcleo del OS 34. En algunas realizaciones, el evaluador 50b de proceso a nivel de núcleo está configurado para interceptar ciertas operaciones del núcleo del OS y, por lo tanto, determinar que el proceso evaluado está intentando realizar ciertas operaciones, que pueden ser maliciosas. Para interceptar tales operaciones, algunas realizaciones pueden emplear un conjunto de mecanismos de filtrado integrados y expuestos por el OS 34. Por ejemplo, en un OS Windows, FltRegisterFilter se puede usar para interceptar operaciones como crear, abrir, escribir y borrar un archivo. En otro ejemplo, el evaluador 50b puede usar ObRegisterCallback para interceptar las operaciones de manejo de objetos crear o duplicar, o PsSetCreateProcessNotifyRoutine para interceptar la creación de nuevos procesos. En otro ejemplo más, las operaciones de registro de Windows, tales como la creación y ajuste de claves/valores de registro, se pueden interceptar usando CmRegisterCallbackEx. Se conocen mecanismos de filtrado similares en la técnica para otros sistemas operativos tales como Linux®. Cuando el evaluador 50b de proceso en modo núcleo intercepta tales operaciones, el evaluador 50b puede formular el indicador 52b de evaluación de proceso y transmitir el indicador 52b al módulo 38 de puntuación de proceso.

Para transmitir datos tales como los indicadores 52a-c de evaluación de proceso desde los evaluadores 50a-c al módulo 38 de puntuación, un experto en la técnica puede emplear cualquier método de comunicación entre procesos. Por ejemplo, para comunicar entre los componentes en modo usuario y en modo núcleo, los evaluadores 50a-c y el módulo 38 se pueden configurar para usar una sección de memoria compartida.

La Figura 8 muestra una secuencia ejemplar de pasos realizados por el motor 40 de introspección de memoria según algunas realizaciones de la presente invención. En un paso 312, el motor 40 puede detectar que un proceso que requiere protección contra programas malignos (en lo sucesivo conocido como proceso protegido) está lanzándose dentro de la VM 32. En algunas realizaciones, tales procesos protegidos incluyen, entre otros, procesos que pertenecen a la aplicación 44 de seguridad.

Para detectar el lanzamiento del proceso protegido, el motor 40 puede emplear estructuras de datos y/o mecanismos nativos para el OS 34. Por ejemplo, algunas versiones del OS Windows® gestionan procesos usando una lista de procesos activos, mantenida por el núcleo. Cada vez que se crea un proceso, se inserta un indicador del proceso respectivo en la lista de procesos activos; el indicador se elimina de la lista tras la terminación del proceso respectivo. En algunas realizaciones, el núcleo del OS 34 representa cada proceso como una estructura de datos, por ejemplo, un bloque de proceso ejecutivo (EPROCESS) en Windows, que comprende, entre otros, el manejo de cada uno de los hilos del proceso respectivo y un ID de proceso único que permite que el OS 34 identifique el proceso respectivo a partir de una pluralidad de procesos en ejecución.

Para detectar la creación del proceso protegido (paso 312 en la Figura 8), algunas realizaciones se enganchan a una función del núcleo que manipula la lista de procesos activos, usando cualquier método de enganche conocido en la técnica. Un ejemplo de tal función del OS Windows es PspInsertProcess, que añade un proceso a la lista de procesos activos cuando el proceso respectivo se lanza a su ejecución. Algunas realizaciones del controlador 36 AM pueden aplicar un parche de redirección a la función del núcleo respectiva, tal como una instrucción VMCALL o una instrucción JMP. Otras realizaciones pueden modificar la entrada de EPT de la función del núcleo respectiva, para

apuntar a una nueva dirección. El efecto de tales parches y/o ganchos de EPT es redirigir la ejecución de la función del OS nativa a un fragmento de código proporcionado por el motor 40 de introspección de memoria. Siguiendo al enganche, cuando el OS 34 intenta lanzar un proceso a su ejecución, el fragmento de código se ejecutará antes o en lugar del código de la función del núcleo respectiva, notificando de este modo al motor 40 de introspección de memoria que el proceso respectivo está ejecutándose. En algunas realizaciones, el motor 40 puede identificar el proceso respectivo según un parámetro (por ejemplo, la estructura EPROCESS que incluye el ID de proceso único) pasado a la función del núcleo cuando se lanza el proceso respectivo. Una realización alternativa puede usar un gancho de memoria (tal como un gancho de EPT) para obtener acceso a una dirección de una sección de memoria que almacena la lista de procesos activos, y según el contenido de la sección de memoria respectiva, determinar además la dirección de la estructura de EPROCESS que describe cada proceso actualmente en ejecución.

En un paso 314, el motor 40 de introspección de memoria puede notificar al controlador 36 AM que el proceso protegido está ejecutándose. Por ejemplo, el motor 40 puede enviar un indicador tal como el ID de proceso del proceso protegido al controlador 36 AM. A continuación, en un paso 316, el motor 40 puede recibir del controlador 36 un indicador de una página de memoria (por ejemplo, una dirección de una página en la memoria virtual), la página de memoria que almacena el código y/o los datos del proceso protegido. En algunas realizaciones, el motor 40 usa los pasos 314-316 para puentear un hueco semántico, que aparece debido a que el motor 40 se ejecuta fuera de la VM 32, mientras que el proceso protegido se ejecuta dentro de la VM 32. El controlador 36 AM, ejecutándose en modo núcleo dentro de la VM 32, puede tener acceso directo a información tal como una dirección de memoria usada por el proceso protegido, por ejemplo, una dirección de una página dentro de la memoria física virtualizada de la VM respectiva (véanse los espacios 114a-b en la Figura 6) almacenando código y/o datos del proceso protegido. Aunque el hipervisor 30 puede obtener acceso a una lista de procesos activos que se ejecutan dentro de la VM respectiva, analizar sintácticamente la lista para determinar todos los módulos (tales como las DLL) cargados por el proceso respectivo y determinar además todas las direcciones de las páginas de memoria que almacenan tales datos/códigos del nivel de hipervisor 30 que pueden requerir un cálculo sustancial. En algunas realizaciones, otra razón para la secuencia de pasos 314-316 es que se pueden intercambiar datos que pertenecen a procesos en modo usuario por el OS 34 entre la memoria 14 física y otros medios legibles por ordenador, por ejemplo, dispositivos 20 de almacenamiento. Ejecutándose fuera de la VM respectiva, el motor 40 de introspección de memoria puede detectar cuándo los datos se intercambian dentro y fuera de la memoria física, pero puede no ser capaz de acceder a y/o proteger tales datos mientras que no residan en la memoria física. Por el contrario, el controlador 36 AM que se ejecuta dentro de la VM 32 puede acceder fácilmente a una página que se intercambia fuera de la memoria física, forzando al OS 34 a cargar la página respectiva. El controlador 36 AM puede enumerar eficientemente de este modo todos los módulos usados/cargados por el proceso protegido, y determinar el tamaño y la ubicación de tales módulos dentro de la memoria física virtualizada de la VM 32.

En una realización alternativa, en lugar de detectar activamente el lanzamiento del proceso protegido (paso 312 anterior), el motor 40 de introspección de memoria puede recibir un indicador del proceso protegido del controlador 36 AM, en donde el controlador 36 AM puede detectar realmente el lanzamiento del proceso protegido desde dentro de la VM 32. En tales realizaciones, el paso 314 como se ha descrito anteriormente ya no es necesario. En otra realización más, en el paso 316, el motor 40 puede realizar realmente los cálculos necesarios para determinar una dirección de la página de memoria del proceso protegido, en lugar de depender del controlador 36 AM como se ha descrito anteriormente.

En un paso 318, el motor de introspección de memoria protege la página objetivo de modificaciones no deseadas, por ejemplo, mediante un programa maligno que intenta comprometer la VM 32. Varios de tales mecanismos de protección de memoria son conocidos en la técnica. La protección se puede forzar por el hipervisor 30 a petición del motor 40 de introspección de memoria, usando estructuras de datos tales como la EPT o la NPT. Por ejemplo, el hipervisor 30 puede establecer la página de memoria objetivo como solamente de lectura, modificando los bits de derecho de acceso a la EPT/NPT de las páginas respectivas. En algunas realizaciones, el hipervisor 30 puede interceptar cualquier intento de escribir en las páginas de memoria asignadas al objeto objetivo y redirigir el intento respectivo al motor 40 de introspección de memoria para su análisis. La operación del motor 40 en el paso 318 se detallará aún más a continuación, en relación con la Figura 9.

Para aplicar protección contra escritura a la página objetivo, el paso 318 puede comprender realizar una traducción de direcciones de memoria del tipo ilustrado en la Figura 6, desde un espacio de memoria virtual configurado por el OS 34 para el proceso protegido, todo el camino hasta el final de la memoria 14 física del sistema 10 central, o desde un espacio de memoria física virtualizada de la VM respectiva a la memoria 14 física. La traducción respectiva permite que el motor 40 de introspección de memoria determine una dirección de la página objetivo en la memoria 14 física real, según el indicador recibido en el paso 316 desde el controlador 36 AM. Tales traducciones pueden emplear un mecanismo de EPT/NPT, como se describe en relación con la Figura 6.

En un paso 320, el motor 40 puede detectar una terminación del proceso protegido. En algunas realizaciones, el paso 320 puede proceder de una manera similar al paso 312 descrito anteriormente. Por ejemplo, el paso 320 puede comprender recibir una señal desde una función de núcleo configurada para eliminar un proceso de la lista de procesos activos de la VM 32, la función respectiva modificada por el controlador 36 AM enganchando (por ejemplo, aplicando un parche, tal como una instrucción VMCALL, a la función respectiva, el parche que redirige la ejecución al motor 40). Una función de Windows ejemplar que se puede modificar de esta forma es PspDeleteProcess.

Cuando el motor 40 detecta la terminación del proceso protegido, un paso 322 elimina la protección de la página objetivo respectiva, por ejemplo, instruyendo al hipervisor 30 para que cambie los permisos de escritura para la página objetivo.

La Figura 9 ilustra una secuencia de pasos realizados por el motor 40 de introspección de memoria para proteger la página objetivo (paso 318 en la Figura 8). En un paso 332, el motor 40 puede interceptar un intento de escribir en la página objetivo; tales intentos pueden ser indicativos de intenciones maliciosas y pueden ser interceptados a través del hipervisor 30, como se ha descrito anteriormente. En un paso 334, el motor 40 puede identificar el proceso que ejecuta el intento; el proceso respectivo se conocerá como el proceso infractor. En algunas realizaciones, para ejecutar el paso 334, el motor 40 puede usar un contenido de un registro de puntero de instrucción, tal como los registros EIP y/o RIP en sistemas x86, para identificar la instrucción del procesador (o la dirección del mismo) que realiza el intento, y un contenido de un registro CR3 para identificar el proceso al que pertenece la instrucción respectiva. Alternativamente, el motor 40 puede usar un contenido de un registro de segmento, tal como los registros FS y GS en procesadores x86, para identificar el proceso infractor según ciertas estructuras de datos del núcleo, que se modifican cada vez que el OS 34 conmuta la ejecución entre procesos.

En un paso 336, el motor 40 puede formular el indicador 52d de evaluación del proceso (véase, por ejemplo, la Figura 4) del proceso infractor y transmitir el indicador 52d al módulo 38 de puntuación de proceso. Un indicador 52d ejemplar puede comprender un indicador (por ejemplo, ID de proceso) del proceso infractor identificado en el paso 334, y un indicador de un tipo de acción intentada por el proceso infractor, e interceptado en el paso 332 (por ejemplo, un intento de escribir en una página de memoria protegida).

Algunos de los métodos y sistemas descritos anteriormente requieren comunicación, tal como intercambio de datos y/o mensajería, entre los componentes que se ejecutan dentro de la VM 32 y los componentes que se ejecutan fuera de la VM respectiva. Tal comunicación se puede llevar a cabo usando cualquier método conocido en la técnica de la virtualización. Por ejemplo, para transmitir datos desde un componente que se ejecuta en modo núcleo, tal como el controlador 36 AM, al motor 40 de introspección de memoria (véase, por ejemplo, el paso 316 en la Figura 8), algunas realizaciones usan una instrucción privilegiada para transferir el control del procesador 12 desde la VM 32 al hipervisor 30. Un ejemplo de tales instrucciones privilegiadas es VMCALL en plataformas Intel, que se puede usar para señalar al motor 40 que algunos datos se están transfiriendo desde dentro de la VM 32. Los datos reales que se transmiten se pueden colocar en una sección de memoria compartida predeterminada entre el controlador 36 y el motor 40. Para transmitir datos desde el motor 40 de introspección de memoria al controlador 36 AM (véase, por ejemplo, el paso 314 en la Figura 8 y el paso 336 en la Figura 9), algunas realizaciones usan un mecanismo de inyección de interrupción para señalar al conductor 36 que los datos se están transmitiendo desde fuera de la VM respectiva. Los datos reales se pueden transferir, por ejemplo, a través de la sección de memoria compartida descrita anteriormente.

En algunas realizaciones, el sistema 10 central se puede configurar para intercambiar información de seguridad, tal como detalles acerca de eventos de detección de programas malignos, con un servidor de seguridad remoto. La Figura 10 ilustra tal configuración ejemplar, en la que una pluralidad de sistemas 10a-c centrales están conectados a un servidor 110 de seguridad a través de una red informática 26. En una realización ejemplar, los sistemas 10a-c centrales son ordenadores individuales usados por empleados de una corporación, mientras que el servidor 110 de seguridad puede comprender un sistema informático configurado por un administrador de red de la corporación respectiva para monitorizar amenazas de programas malignos o eventos de seguridad que ocurren en los sistemas 10a-c. En otra realización, por ejemplo, en un sistema de Infraestructura como servicio (IAAS) en donde cada sistema 10a-c central es un servidor que aloja decenas o centenas de máquinas virtuales, el servidor 110 de seguridad puede comprender un sistema informático configurado para gestionar operaciones anti-programa maligno para todas de tales VM desde una ubicación central. En otra realización más, el servidor 110 de seguridad puede comprender un sistema informático configurado por un proveedor de software anti-programa maligno (por ejemplo, el proveedor de la aplicación 44 de seguridad, entre otros), para recibir datos estadísticos y/o de comportamiento acerca del programa maligno detectado en varios sistemas alrededor de la red 26. La red 26 puede incluir una red de área extensa tal como Internet, mientras que partes de la red 26 pueden incluir redes de área local (LAN).

La Figura 11 muestra un intercambio de datos ejemplar entre el sistema 10 central y el servidor 110 de seguridad en una realización como se muestra en la Figura 10. El sistema 10 central se puede configurar para enviar un informe 80 de seguridad al servidor 110 y recibir un conjunto de ajustes 82 de seguridad desde el servidor 110. En algunas realizaciones, el informe 80 de seguridad comprende indicadores y/o puntuaciones de evaluación de procesos determinados por evaluadores de procesos que se ejecutan en el sistema 10 central, y/o puntuaciones agregadas determinadas por el módulo 38 de puntuación de proceso, entre otros. El informe 80 de seguridad también puede comprender datos que identifiquen la máquina virtual y los procesos evaluados respectivos (por ejemplo, ID de procesos, nombres, rutas, comprobaciones aleatorias, información de versión u otro tipo de identificadores de aplicaciones y/o procesos), así como indicadores que asocian un indicador/puntuación de evaluación de proceso a la VM y al proceso para el cual se determinó. En algunas realizaciones, el informe 80 puede comprender además datos estadísticos y/o de comportamiento con respecto a procesos y/o aplicaciones que se ejecutan en el sistema 10 central. El sistema 10 se puede configurar para enviar el informe 80 tras la detección del programa maligno y/o según una programación (por ejemplo, cada pocos minutos, cada hora, etc.).

En algunas realizaciones, los ajustes 82 de seguridad pueden incluir parámetros operativos de evaluadores de procesos (por ejemplo, parámetros de filtros 50a-c en la Figura 4), y/o parámetros del módulo 38 de puntuación de proceso. Un ejemplo de un parámetro operativo del módulo 38 es el umbral para decidir si un proceso evaluado es malicioso (véase el paso 308 en la Figura 5 y la descripción asociada). Un parámetro operativo ejemplar de un evaluador de proceso es un valor de una puntuación de malicia asignado a un proceso evaluado, cuando el proceso evaluado realiza una cierta acción. Por ejemplo, un proceso evaluado puede recibir una puntuación de malicia de 0.1 cuando el proceso respectivo escribe en un archivo de disco, y una puntuación de malicia de 0.7 cuando modifica un valor de registro de Windows.

En algunas realizaciones, el servidor 110 ejecuta un algoritmo de optimización para ajustar dinámicamente tales parámetros para maximizar el rendimiento de detección de programas malignos, por ejemplo, para aumentar la tasa de detección mientras se minimizan los falsos positivos. Los algoritmos de optimización pueden recibir datos estadísticos y/o de comportamiento acerca de varios procesos que se ejecutan en la pluralidad de sistemas 10a-c centrales, incluyendo indicadores/puntuaciones de evaluación de procesos reportados al módulo 38 de puntuación de proceso por varios evaluadores de procesos, y determinar valores óptimos para los parámetros. Los valores se transmiten entonces a los sistemas centrales respectivos a través de la red 26. En algunas realizaciones, para determinar el valor de parámetro óptimo, el servidor 110 puede calibrar la operación del módulo 38 de puntuación de proceso y/o los evaluadores 50a-c de proceso usando un conjunto de procesos que se sabe que están limpios (no afectados por programas malignos). En un escenario de calibración ejemplar, el servidor 110 de seguridad puede instruir al sistema 10 central para ejecutar un conjunto de procesos de calibración, que se sabe que están limpios, y enviar de vuelta al servidor 110 un conjunto de indicadores/puntuaciones de evaluación de procesos determinados para los procesos de calibración. El servidor 110 puede determinar entonces los valores de parámetros adaptados a la máquina virtual y/o al sistema central respectivos.

En otro ejemplo, los ajustes 82 de seguridad comprenden un conjunto de valores de ponderación usados por el módulo 38 de puntuación de proceso para determinar una puntuación de malicia agregada para un proceso evaluado según los indicadores de evaluación de proceso individuales recibidos desde varios evaluadores de proceso. En una realización en donde la puntuación agregada es una suma ponderada o una media ponderada de puntuaciones individuales, y en donde cada puntuación se calcula según un criterio o método de detección de programa maligno distinto (por ejemplo, cuando cada puntuación indica si un proceso evaluado realiza un cierto comportamiento indicativo de programa maligno), cambiar la ponderación de una puntuación individual puede cambiar efectivamente la relevancia del criterio o método respectivo, en comparación con otros criterios/métodos. Las amenazas de programa maligno típicamente ocurren en oleadas, en las que un gran número de sistemas informáticos en todo el mundo se ven afectados por el mismo agente de programa maligno en un corto intervalo de tiempo. Recibiendo los informes 80 de seguridad en tiempo real desde una pluralidad de sistemas centrales, el servidor 110 de seguridad se puede mantener actualizado con las amenazas de programas malignos actuales y puede entregar rápidamente ajustes 82 de seguridad óptimos a los sistemas centrales respectivos, los ajustes 82 que incluyen, por ejemplo, un conjunto de ponderaciones optimizadas para detectar las amenazas de programas malignos actuales.

Los sistemas y métodos ejemplares descritos anteriormente permiten proteger un sistema central, tal como un sistema informático, del programa maligno tal como virus y herramientas de sustitución de ficheros del sistema raíz. Los sistemas anti-programas malignos convencionales típicamente se ejecutan en el nivel de privilegio del procesador del sistema operativo (por ejemplo, modo núcleo). Algunos programas malignos, tales como las herramientas de sustitución de ficheros del sistema raíz, también pueden operar en el nivel del OS y, de este modo, pueden incapacitar los sistemas anti-programas malignos convencionales y comprometer la seguridad del sistema informático. Por el contrario, en algunas realizaciones de la presente invención, un hipervisor se ejecuta en el sistema informático en el nivel de privilegio de procesador más alto, desplazando el sistema operativo a una máquina virtual. Un sistema anti-programa maligno que opera según algunas realizaciones de la presente invención comprende componentes que se ejecutan dentro de la VM y componentes que se ejecutan fuera de la VM, en el nivel de hipervisor. Algunas operaciones anti-programa maligno se pueden dirigir, de este modo, desde un nivel de privilegio de procesador más alto que el del sistema operativo, donde no se pueden trastocar por un programa maligno que se ejecuta dentro de la VM. En algunas realizaciones, un único motor de introspección de memoria, que se ejecuta en el nivel de hipervisor, puede proteger múltiples máquinas virtuales que se ejecutan concurrentemente en el sistema informático respectivo.

En algunas realizaciones, la operación del motor de introspección de memoria incluye seleccionar un conjunto de objetos de software críticos, tales como ciertos controladores, bibliotecas, registros y tablas de páginas, entre otros, y evitar cambios maliciosos en tales objetos. En particular, algunas realizaciones pueden proteger de este modo los componentes anti-programas malignos que se ejecutan dentro de la VM de ataques maliciosos.

Para proteger tales objetos, algunas realizaciones pueden evitar cambios maliciosos interceptando un intento de escribir en un espacio de memoria asignado al objeto respectivo y bloqueando o redirigiendo el intento. Otras realizaciones pueden proteger un objeto objetivo marcando el espacio de memoria asignado al objeto respectivo como de sólo lectura. En configuraciones típicas de hardware y software, la memoria se divide en bloques individuales de direcciones contiguas, conocidos como páginas. En sistemas que soportan virtualización, los permisos de acceso a las páginas se controlan por el hipervisor, por ejemplo, usando estructuras de datos

dedicadas, tales como tablas de páginas extendidas (EPT) en plataformas Intel. De este modo, la protección del espacio de memoria de un objeto objetivo se puede lograr, por ejemplo, mediante un motor de introspección de memoria que instruye al hipervisor para que marque un conjunto de páginas que contienen datos que pertenecen al objeto respectivo como de sólo lectura.

En algunas realizaciones, algunos componentes anti-programa maligno se ejecutan dentro de la máquina virtual protegida, colaborando con el motor de introspección de memoria para detectar el programa maligno. Tales configuraciones pueden simplificar sustancialmente la detección del programa maligno, puenteando un hueco semántico que surge a través de la virtualización. En configuraciones de software típicas, un componente de detección de programa maligno que se ejecuta en modo de usuario puede tener acceso a una riqueza de información acerca del comportamiento de un proceso evaluado, mientras que la mayor parte de esta información no está fácilmente disponible para los componentes que se ejecutan a nivel de núcleo o fuera de la VM respectiva. Por ejemplo, cuando un proceso evaluado intenta descargar un archivo de Internet, un evaluador de proceso en modo usuario, por ejemplo, usando métodos conocidos en la técnica, tales como inyección de DLL, puede identificar qué proceso está realizando la acción, puede detectar que el proceso evaluado está intentando descargar un archivo, y puede determinar la dirección IP desde la que se descarga el archivo, y la ubicación en el disco del archivo descargado, entre otros. Mientras tanto, un evaluador de proceso que se ejecute a nivel de hipervisor solamente puede detectar que un conjunto de paquetes de red está circulando sobre un adaptador de red del sistema central. Aunque puede ser posible, en principio, recuperar información acerca del comportamiento del proceso evaluado desde el nivel del hipervisor, tales tareas pueden ser poco prácticas para la detección de programa maligno, dado que pueden conllevar un coste computacional significativo. Combinando componentes anti-programa maligno que se ejecutan dentro de la VM respectiva con un motor de introspección de memoria que se ejecuta fuera de la VM, algunas realizaciones de la presente invención pueden usar la abundancia de datos de comportamiento a los que tienen acceso los componentes dentro de la VM, mientras se protege la integridad de tales componentes desde fuera de la VM respectiva.

En los sistemas anti-programas malignos convencionales, un componente de software, que se ejecuta en un nivel de privilegio de procesador similar al del sistema operativo, detecta cuándo se está lanzando un proceso e instruye a otros componentes anti-programas malignos para monitorizar el comportamiento del proceso respectivo. Algunos agentes de programa maligno logran comprometer tales sistemas anti-programas malignos deshabilitando el componente de software que detecta los lanzamientos de procesos, haciendo de este modo que el sistema anti-programa maligno monitorice solamente un subconjunto de los procesos que se ejecutan actualmente. Por el contrario, en algunas realizaciones de la presente invención, el componente que detecta los lanzamientos de procesos se mueve fuera de la máquina virtual respectiva, a un nivel de privilegio de procesador más alto que el sistema operativo. Tales configuraciones pueden evitar que el programa maligno se oculte de los componentes anti-programas malignos.

En algunas realizaciones, un módulo de puntuación de proceso recibe indicadores de evaluación por proceso desde una pluralidad de evaluadores de proceso que se ejecutan o bien dentro o bien fuera de la VM respectiva. Los indicadores de evaluación de procesos recibidos desde los componentes que se ejecutan dentro de la VM protegida pueden indicar, por ejemplo, que un proceso evaluado ha realizado un comportamiento indicativo de programa maligno, tal como un intento de modificar un valor de registro del OS, o un intento de eliminar un archivo. Los indicadores de evaluación de proceso determinados fuera de la VM respectiva pueden indicar, por ejemplo, que un proceso evaluado está intentando escribir en una sección de memoria protegida. Los indicadores de evaluación de proceso pueden comprender puntuaciones numéricas que indican un grado de malicia del proceso respectivo. En algunas realizaciones, el módulo de puntuación de proceso determina una puntuación agregada según la pluralidad de indicadores/puntuaciones de evaluación de proceso recibidos desde varios evaluadores de proceso, y determina si el proceso evaluado es malicioso según la puntuación agregada.

Para un experto en la técnica, estará claro que las realizaciones anteriores se pueden alterar de muchas formas sin apartarse del alcance de la invención. Por consiguiente, el alcance de la invención se debería determinar por las siguientes reivindicaciones.

REIVINDICACIONES

1. Un sistema [10] central que comprende al menos un procesador [12] hardware configurado para ejecutar:

un hipervisor [30] configurado para exponer una máquina virtual [32], el hipervisor que se ejecuta en un mayor nivel de privilegio de procesador que el software que se ejecuta dentro de la máquina virtual;

un evaluador [50] de proceso que se ejecuta dentro de la máquina virtual [32];

un motor [40] de introspección de memoria que se ejecuta fuera de la máquina virtual [32] en un nivel de privilegio de procesador del hipervisor; y

un módulo [38] de puntuación de proceso,

en donde el evaluador [50] de proceso está configurado para:

determinar si un proceso evaluado que se ejecuta dentro de la máquina virtual [32] realiza una acción, y

en respuesta, cuando el proceso evaluado realiza la acción, transmitir un primer indicador de evaluación de proceso al módulo [38] de puntuación de proceso, el primer indicador de evaluación de proceso determinado para el proceso evaluado;

el motor [40] de introspección de memoria configurado para:

interceptar una llamada a una función del sistema operativo, para detectar el lanzamiento de un proceso protegido que se ejecuta dentro de la máquina virtual [32], en donde la función del sistema operativo está configurada para añadir el proceso protegido a una lista de procesos que se ejecutan dentro de la máquina virtual [32], y

en respuesta a la detección del lanzamiento,

determinar si el proceso evaluado intenta modificar una página de memoria del proceso protegido y,

en respuesta, cuando el proceso evaluado intenta modificar la página de memoria,

transmitir un segundo indicador de evaluación de proceso al módulo [38] de puntuación de proceso, el segundo indicador de evaluación de proceso determinado para el proceso evaluado;

y el módulo [38] de puntuación de proceso configurado para:

recibir el primer y segundo indicadores de evaluación de proceso, y

en respuesta, determinar si el proceso evaluado es malicioso según el primer y segundo indicadores de evaluación de proceso.

2. El sistema central de la reivindicación 1, en donde el motor [40] de introspección de memoria está configurado además para:

en respuesta a la detección del lanzamiento del proceso protegido, enviar un indicador del proceso protegido a una aplicación [44] de seguridad que se ejecuta dentro de la máquina virtual, y

en respuesta, recibir desde la aplicación [44] de seguridad un indicador de la página de memoria.

3. El sistema central de la reivindicación 1, en donde el evaluador [50] de proceso comprende un evaluador [50a] de proceso a nivel de usuario que se ejecuta a nivel de usuario de privilegio de procesador, el evaluador [50a] de proceso a nivel de usuario configurado para determinar si el proceso evaluado realiza la acción.

4. El sistema central de la reivindicación 1, en donde el evaluador [50] de proceso comprende un evaluador [50b] de proceso a nivel de núcleo que se ejecuta a nivel de núcleo de privilegio de procesador, el evaluador [50b] de proceso a nivel de núcleo configurado para determinar si el proceso evaluado realiza la acción.

5. El sistema central de la reivindicación 1, en donde el evaluador [50] de proceso comprende un evaluador [50c] de llamadas al sistema configurado para interceptar una llamada al sistema realizada por el proceso evaluado.

6. El sistema central de la reivindicación 1, en donde el módulo [38] de puntuación de proceso se ejecuta dentro de la máquina virtual [32].

7. El sistema central de la reivindicación 1, en donde el módulo [38] de puntuación de proceso se ejecuta fuera de la máquina virtual [32].

8. El sistema central de la reivindicación 1, en donde determinar si el proceso evaluado es malicioso comprende determinar una puntuación agregada según una primera puntuación y una segunda puntuación, la primera y

segunda puntuaciones determinadas en un servidor [110] de seguridad según el primer y segundo indicadores de evaluación de proceso, respectivamente, en donde el servidor de seguridad [110] está configurado para realizar transacciones anti-programas malignos con una pluralidad de sistemas informáticos, incluyendo el sistema [10] central.

9. El sistema central de la reivindicación 1, en donde determinar si el proceso evaluado es malicioso comprende determinar una puntuación agregada según una suma ponderada de una primera puntuación y una segunda puntuación, la primera y segunda puntuaciones determinadas según el primer y segundo indicadores de evaluación de proceso, respectivamente, y en donde el módulo [38] de puntuación de proceso está configurado además para recibir desde un servidor [110] de seguridad una primera ponderación y una segunda ponderación, la primera ponderación que multiplica la primera puntuación en la suma ponderada, y la segunda ponderación que multiplica la segunda puntuación en la suma ponderada, y en donde el servidor [110] de seguridad está configurado para realizar transacciones anti-programas malignos con una pluralidad de sistemas informáticos, incluyendo el sistema [10] central.

10. El sistema central de la reivindicación 1, en donde el proceso protegido incluye el módulo [38] de puntuación de proceso.

11. El sistema central de la reivindicación 1, en donde el proceso protegido forma parte de una aplicación [44] de seguridad que comprende el evaluador [50] de proceso.

12. Un medio legible por ordenador no transitorio que codifica instrucciones que, cuando se ejecutan en un sistema [10] central que comprende al menos un procesador [12], hacen que el sistema [10] central forme:

un evaluador [50] de proceso que se ejecuta dentro de una máquina virtual [32] expuesta por un hipervisor [40] que se ejecuta en el sistema central, el hipervisor que tiene un mayor privilegio de procesador que el software que se ejecuta dentro de la máquina virtual;

un motor [40] de introspección de memoria que se ejecuta fuera de la máquina virtual [32] en un nivel de privilegio de procesador del hipervisor; y

un módulo [38] de puntuación de proceso,

en donde el evaluador [50] de proceso está configurado para:

determinar si un proceso evaluado que se ejecuta dentro de la máquina virtual [32] realiza una acción y,

en respuesta, cuando el proceso evaluado realiza la acción, transmitir un primer indicador de evaluación de proceso al módulo [38] de puntuación de proceso, el primer indicador de evaluación de proceso determinado para el proceso evaluado;

el motor [40] de introspección de memoria configurado para:

interceptar una llamada a una función del sistema operativo, para detectar el lanzamiento de un proceso protegido que se ejecuta dentro de la máquina virtual [32], en donde la función del sistema operativo se ejecuta dentro de la máquina virtual [32] y está configurada para añadir el proceso protegido a una lista de procesos que se ejecutan dentro de la máquina virtual [32], y

en respuesta a la detección del lanzamiento,

determinar si el proceso evaluado intenta modificar una página de memoria del proceso protegido, y

en respuesta, cuando el proceso evaluado intenta modificar la página de memoria, transmitir un segundo indicador de evaluación de proceso al módulo [38] de puntuación de proceso, el segundo indicador de evaluación de proceso determinado para el proceso evaluado; y

el módulo [38] de puntuación de proceso configurado para:

recibir el primer y segundo indicadores de evaluación de proceso, y

en respuesta, determinar si el proceso evaluado es malicioso según el primer y segundo indicadores de evaluación de proceso.

13. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el motor [40] de introspección de memoria está configurado además para:

en respuesta a la detección del lanzamiento del proceso protegido, enviar un indicador del proceso protegido a una aplicación [44] de seguridad que se ejecuta dentro de la máquina virtual [32], y

en respuesta, recibir desde la aplicación [44] de seguridad un indicador de la página de memoria.

14. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el evaluador [50] de proceso comprende un evaluador [50a] de proceso a nivel de usuario que se ejecuta a nivel de usuario de privilegio de procesador, el evaluador [50a] de proceso a nivel de usuario configurado para determinar si el proceso evaluado realiza la acción.

15. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el evaluador [50] de proceso comprende un evaluador [50b] de proceso a nivel de núcleo que se ejecuta a nivel de núcleo de privilegio de procesador, el evaluador [50b] de proceso a nivel de núcleo configurado para determinar si el proceso evaluado realiza la acción.

16. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el evaluador [50] de proceso comprende un evaluador [50c] de llamada al sistema configurado para interceptar una llamada al sistema realizada por el proceso evaluado.

17. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el módulo [38] de puntuación de proceso se ejecuta dentro de la máquina virtual.

18. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el módulo [38] de puntuación de proceso se ejecuta fuera de la máquina virtual.

19. El medio legible por ordenador no transitorio de la reivindicación 12, en donde determinar si el proceso evaluado es malicioso comprende determinar una puntuación agregada según una primera puntuación y una segunda puntuación, la primera y segunda puntuaciones determinadas en un servidor [110] de seguridad según el primer y segundo indicadores de evaluación de proceso, respectivamente, en donde el servidor [110] de seguridad está configurado para realizar transacciones anti-programas malignos con una pluralidad de sistemas informáticos, incluyendo el sistema [10] central.

20. El medio legible por ordenador no transitorio de la reivindicación 12, en donde determinar si el proceso evaluado es malicioso comprende determinar una puntuación agregada según una suma ponderada de una primera puntuación y una segunda puntuación, la primera y segunda puntuaciones determinadas según el primer y segundo indicadores de evaluación de proceso, respectivamente, y en donde el módulo [38] de puntuación de proceso está configurado además para recibir desde un servidor de seguridad [110] una primera ponderación y una segunda ponderación, la primera ponderación que multiplica la primera puntuación en la suma ponderada, y la segunda ponderación que multiplica la segunda puntuación en la suma ponderada, y en donde el servidor [110] de seguridad está configurado para realizar transacciones anti-programas malignos con una pluralidad de sistemas informáticos, incluyendo el sistema [10] central.

21. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el proceso protegido incluye el módulo [38] de puntuación de proceso.

22. El medio legible por ordenador no transitorio de la reivindicación 12, en donde el proceso protegido forma parte de una aplicación [44] de seguridad configurada para ejecutar el evaluador [50] de proceso.

23. Un método que comprende:

emplear al menos un procesador [12] de un sistema [10] central para ejecutar un motor [40] de introspección de memoria, un evaluador [50] de proceso y un módulo [38] de puntuación, el motor [40] de introspección de memoria que se ejecuta fuera de una máquina virtual [32] expuesta por un hipervisor [30] que se ejecuta en el sistema [10] central, el evaluador [50] de proceso que se ejecuta dentro de la máquina virtual [32], en donde el hipervisor tiene un mayor privilegio de procesador que el software que se ejecuta dentro de la máquina virtual, en donde el motor de introspección de memoria se ejecuta en un nivel de privilegio de procesador del hipervisor, y en donde el evaluador [50] de proceso está configurado para:

determinar si un proceso evaluado que se ejecuta dentro de la máquina virtual [32] realiza una acción y,

en respuesta, cuando el proceso evaluado realiza la acción, transmitir un primer indicador de evaluación de proceso al módulo [38] de puntuación de proceso, el primer indicador de evaluación de proceso determinado para el proceso evaluado, emplear el motor [40] de introspección de memoria para:

interceptar una llamada a una función del sistema operativo para detectar el lanzamiento de un proceso protegido que se ejecuta dentro de la máquina virtual [32], en donde la función del sistema operativo está configurada para añadir el proceso protegido a una lista de procesos que se ejecutan dentro de la máquina virtual [32], y

en respuesta a la detección del lanzamiento:

determinar si el proceso evaluado intenta modificar una página de memoria del proceso protegido, y

en respuesta, cuando el proceso evaluado intenta modificar la página de memoria, transmitir un segundo indicador de evaluación de proceso al módulo [38] de puntuación de proceso, el segundo indicador de evaluación de proceso determinado para el proceso evaluado;

emplear el módulo de puntuación [38] para:

recibir el primer y segundo indicadores de evaluación de proceso y,

en respuesta, determinar si el proceso evaluado es malicioso según el primer y segundo indicadores de evaluación de proceso.

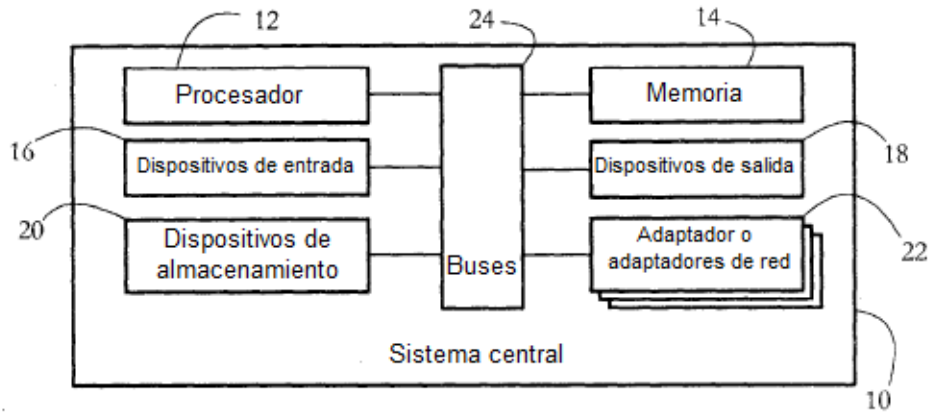
12
24
14
Procesador
Memoria
16
Dispositivos de entrada
Dispositivos de salida
18
20
Dispositivos de almacenamiento
Buses
Adaptador o adaptadores de red
22
Sistema central
10

FIG. 1

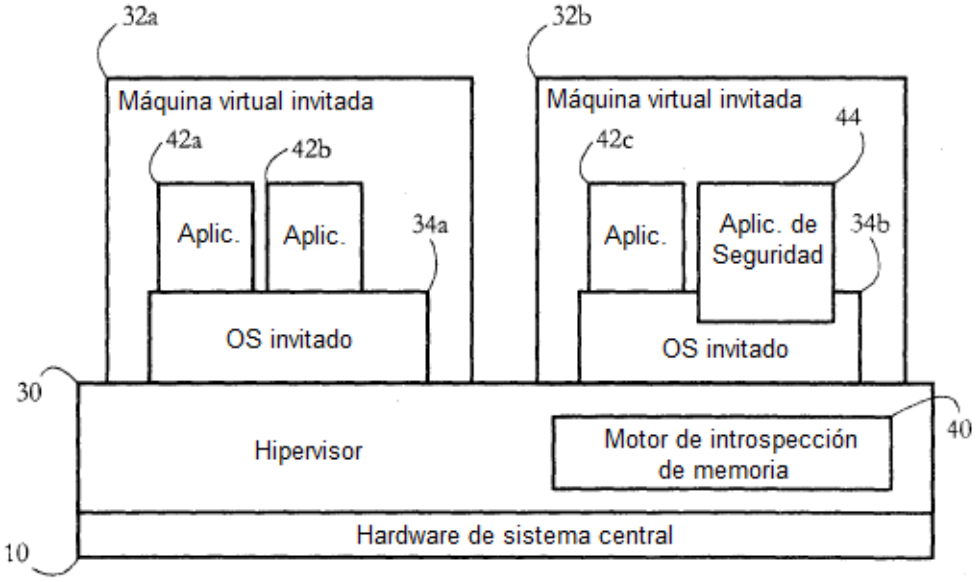
32a
32b
Máquina virtual invitada
Máquina virtual invitada
42a
42b
42c
44
Aplic.
Aplic.
34a
Aplic.
Aplic. de Seguridad
34b
OS invitado
OS invitado
30
Hipervisor
Motor de introspección de memoria
40
Hardware de sistema central
10

FIG. 2

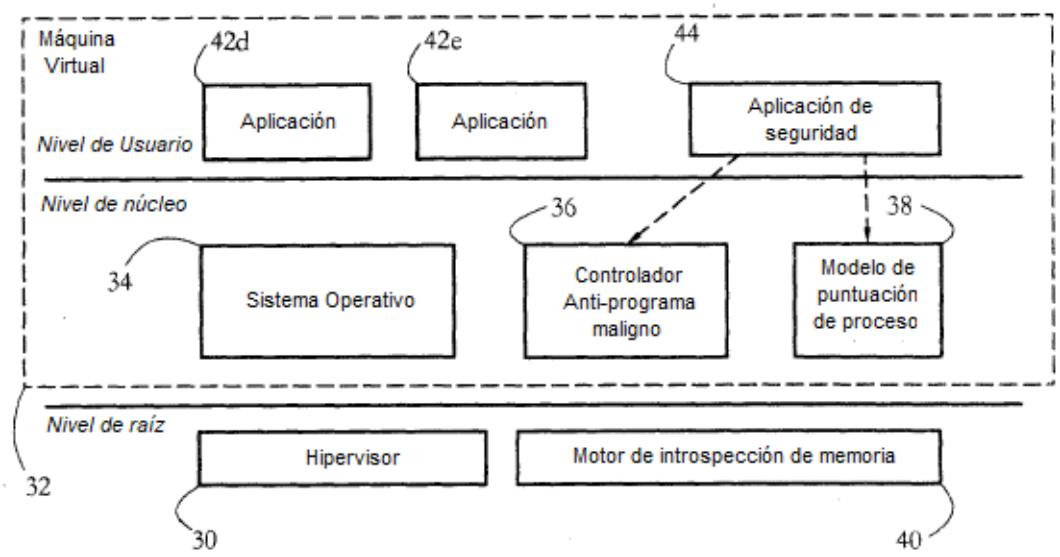
Máquina Virtual
42d
42e
44
Aplicación
Aplicación
Aplicación de seguridad
Nivel de Usuario
Nivel de núcleo
36
38
34
Sistema Operativo
Controlador Anti-programa maligno
Modelo de puntuación de proceso
Nivel de raíz
Hipervisor
Motor de introspección de memoria
32
30
40

FIG. 3

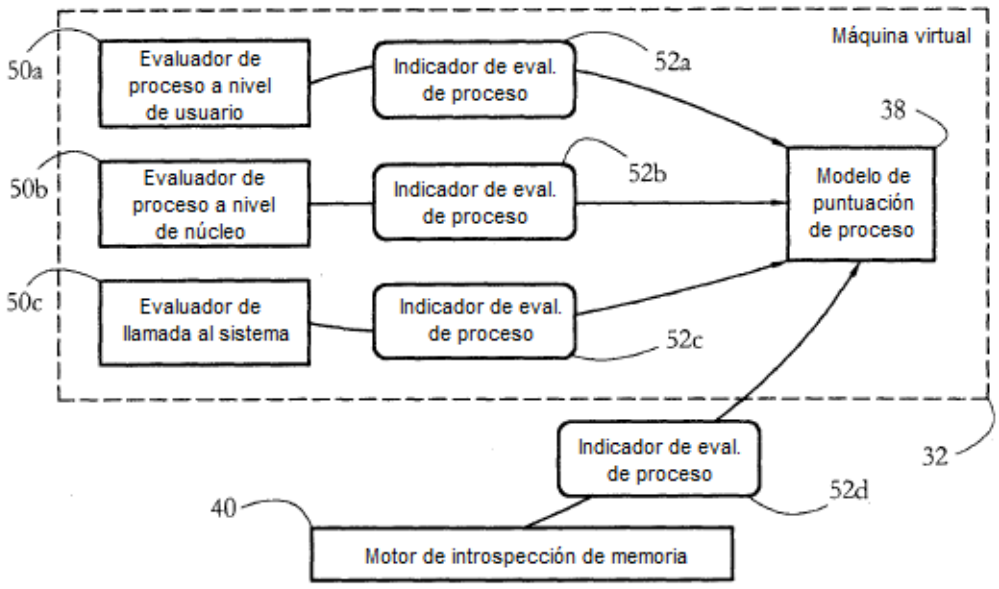
Máquina virtual
50a
Evaluador de proceso a nivel de usuario
Indicador de eval. de proceso
52a
38
50b
Evaluador de proceso a nivel de núcleo
Indicador de eval. de proceso
52b
Modelo de puntuación de proceso
50c
Evaluador de llamada al sistema
Indicador de eval. de proceso
52c
32
Indicador de eval. de proceso
52d
40
Motor de introspección de memoria

FIG. 4

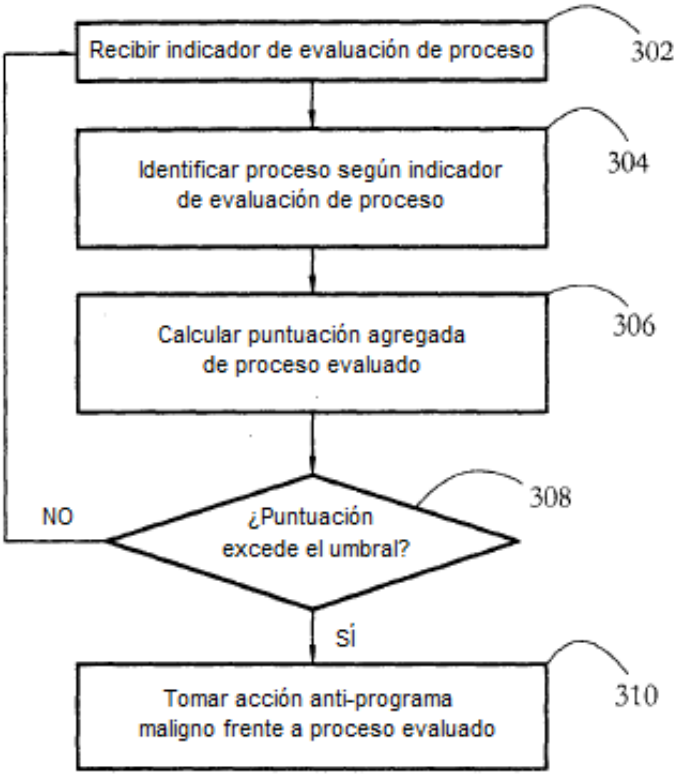
Recibir indicador de evaluación de proceso
302
Identificar proceso según indicador de evaluación de proceso
304
Calcular puntuación agregada de proceso evaluado
306
¿Puntuación excede el umbral?
308
NO
SÍ
Tomar acción anti-programa maligno frente a proceso evaluado
310

FIG. 5

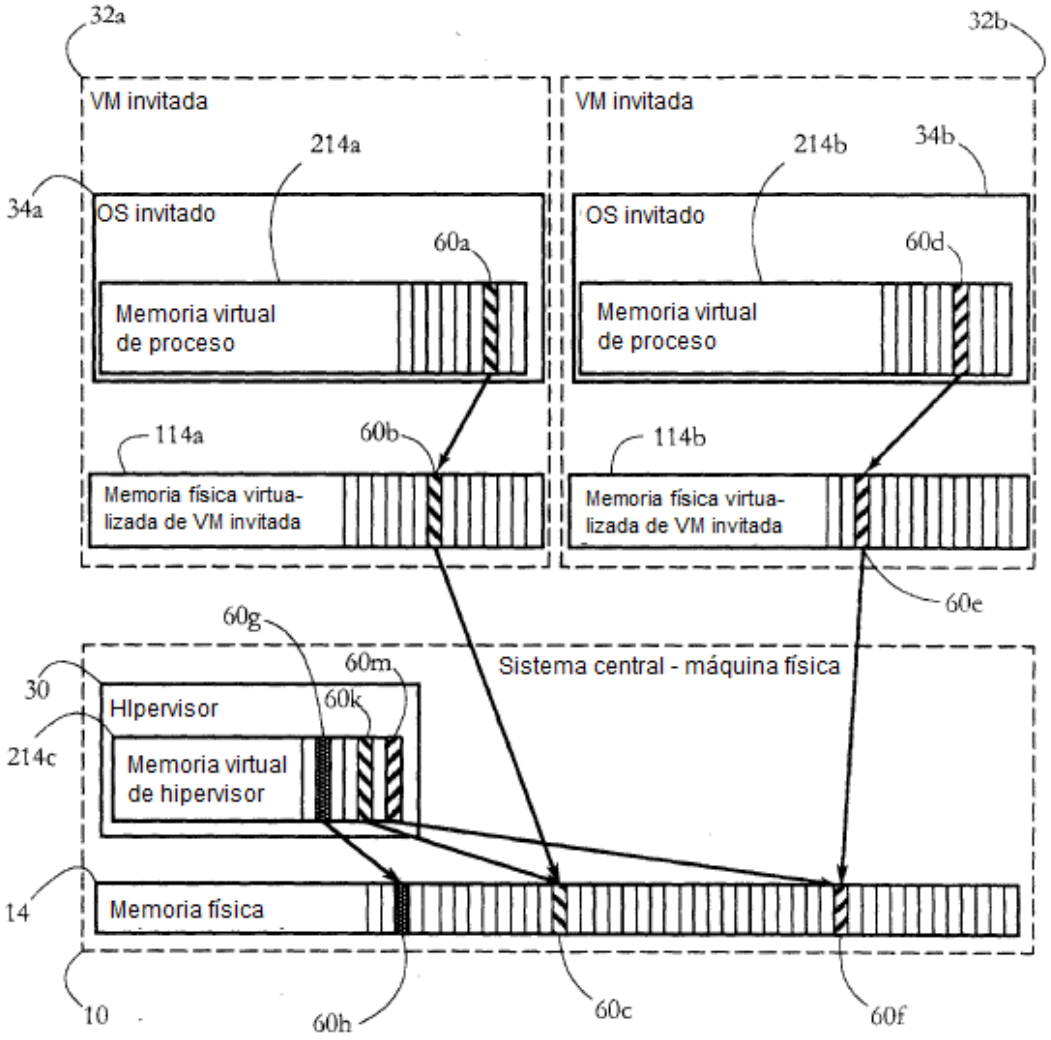
32a
32b
VM invitada
VM invitada
214a
214b
34b
34a
OS invitado
OS invitado
60a
60d
Memoria virtual de proceso
Memoria virtual de proceso
114a
60b
114b
Memoria física virtua-lizada de VM invitada
Memoria física virtua-lizada de VM invitada
60e
60g
60m
Sistema central - máquina física
30
Hipervisor
60k
214c
Memoria virtual de hipervisor
14
Memoria física
10
60h
60c
60f

FIG. 6

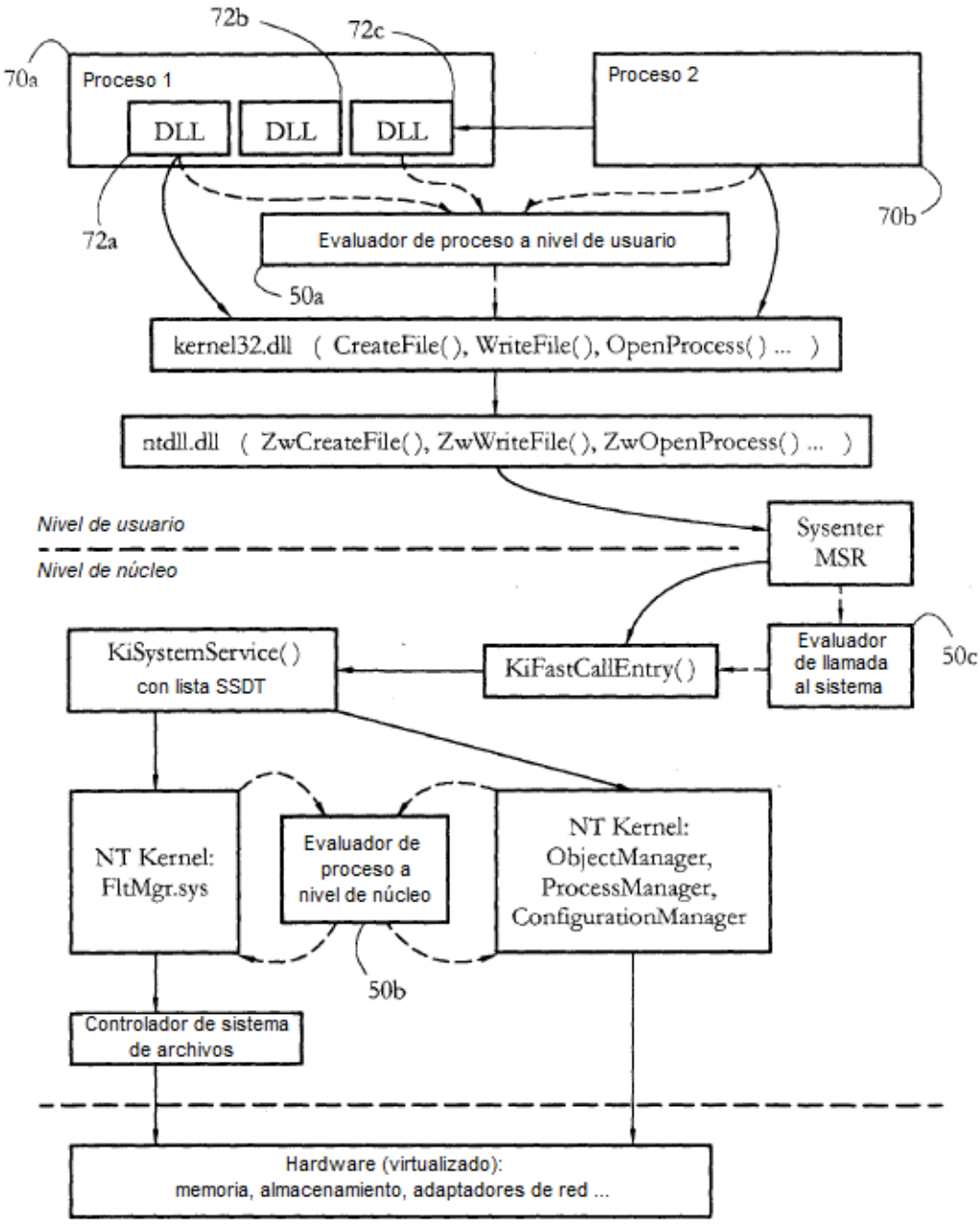
72b
72c
70a
Proceso 1
DLL
DLL
DLL
Proceso 2
70b
72a
Evaluador de proceso a nivel de usuario
50a
kernel32.dll (CreateFile( ), WriteFile( ), OpenProcess( ) ...)
ntdll.dll (ZwCreateFile( ), ZwWriteFile( ), ZwOpenProcess( ) ...)
Nivel de usuario
Nivel de núcleo
Sysenter
MSR
KiSystemService( )
con lista SSDT
KiFastCallEntry( )
Evaluador
de llamada
al sistema
50c
NT Kernel:
FltMgr.sys
Evaluador de
proceso a
nivel de núcleo
NT Kernel:
ObjectManager,
ProcessManager,
ConfigurationManager
50b
Controlador de sistema
de archivos
Hardware (virtualizado):
memoria, almacenamiento, adaptadores de red ...

FIG. 7

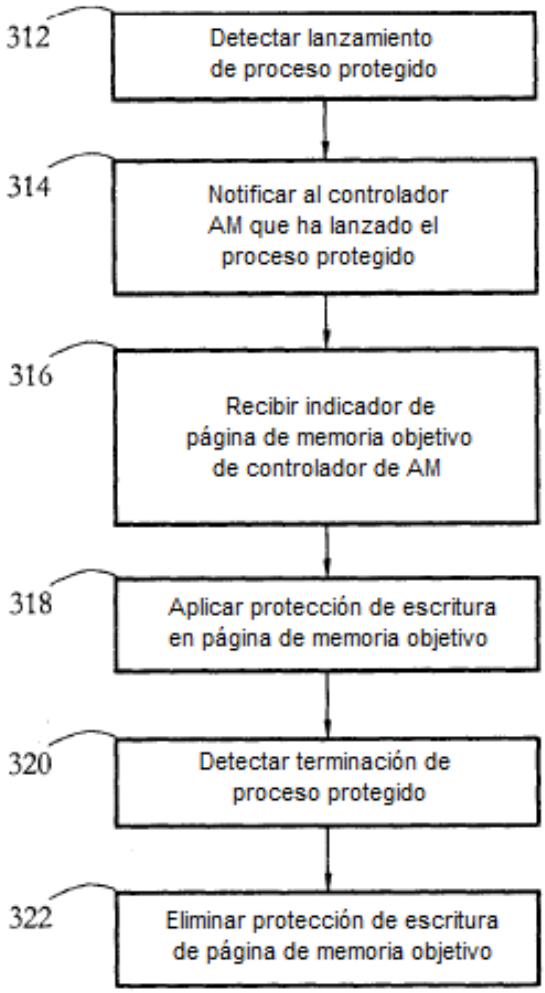
312
Detectar lanzamiento de proceso protegido
314
Notificar al controlador AM que ha lanzado el proceso protegido
316
Recibir indicador de página de memoria objetivo de controlador de AM
318
Aplicar protección de escritura en página de memoria objetivo
320
Detectar terminación de proceso protegido
322
Eliminar protección de escritura de página de memoria objetivo

FIG. 8

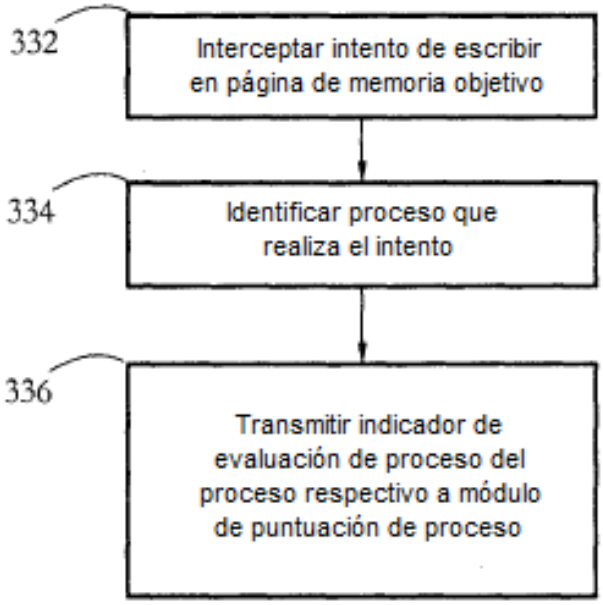
332
Interceptar intento de escribir en página de memoria objetivo
334
Identificar proceso que realiza el intento
336
Transmitir indicador de evaluación de proceso del proceso respectivo a módulo de puntuación de proceso

FIG. 9

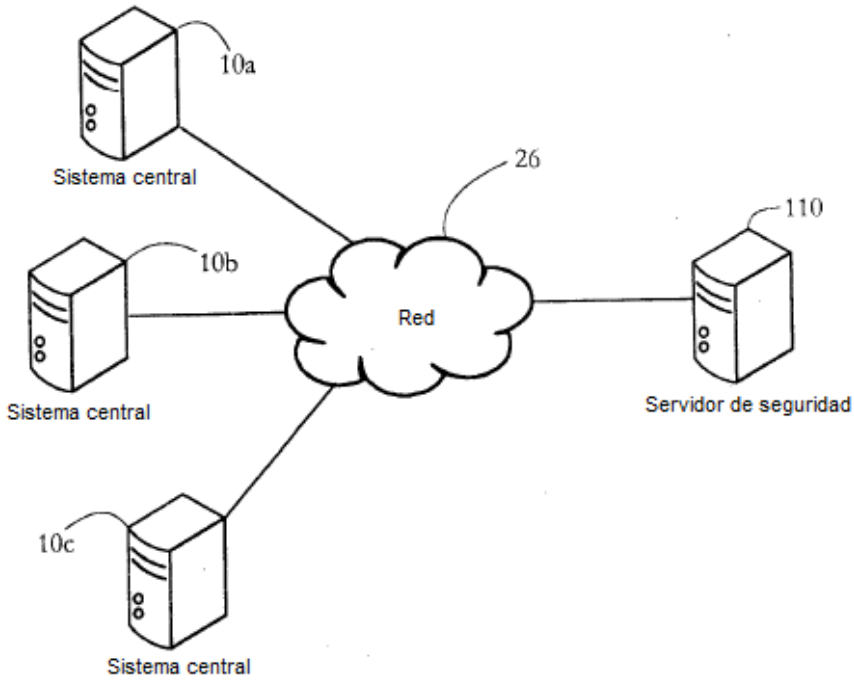
10a
Sistema central
26
110
10b
Red
Sistema central
Servidor de seguridad
10c
Sistema central

FIG. 10

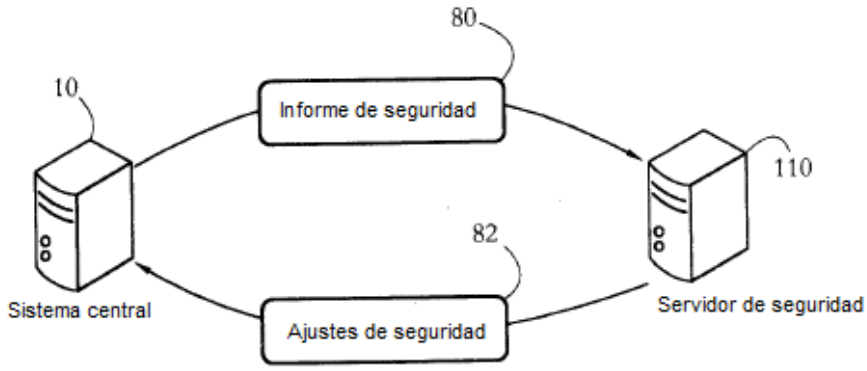
80
10
Informe de seguridad
110
82
Sistema central
Ajustes de seguridad
Servidor de seguridad

FIG. 11