

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 372 989**

51 Int. Cl.:
G06F 21/00

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **08003059 .6**
96 Fecha de presentación: **14.07.2004**
97 Número de publicación de la solicitud: **1916614**
97 Fecha de publicación de la solicitud: **30.04.2008**

54 Título: **PROYECCIÓN DE FIABILIDAD DESDE UN ENTORNO DE CONFIANZA A UN ENTORNO SIN CONFIANZA.**

30 Prioridad:
07.08.2003 US 638199

45 Fecha de publicación de la mención BOPI:
30.01.2012

45 Fecha de la publicación del folleto de la patente:
30.01.2012

73 Titular/es:
**MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WASHINGTON 98052-6399, US**

72 Inventor/es:
**Willman, Bryan Mark;
England, Paul;
Ray, Kenneth D.;
Kaplan, Keith;
Kurien, Varugis y
Marr, Michael David**

74 Agente: **Carpintero López, Mario**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

ES 2 372 989 T3

DESCRIPCIÓN

Proyección de fiabilidad desde un entorno de confianza a un entorno sin confianza

Campo de la invención

- 5 La presente invención se refiere, en general al campo de la seguridad de los ordenadores. Más en particular, esta invención versa acerca del uso de entornos plurales de ejecución (por ejemplo, sistemas operativos) en un único dispositivo informático, y proporciona técnicas que soportan la fiabilidad de tales sistemas o entornos operativos.

Antecedentes de la invención

- 10 Los primeros ordenadores solo eran capaces de ejecutar un único programa a la vez. Sin embargo, en los tiempos modernos se espera de los ordenadores que sean capaces de ejecutar varios soportes lógicos diferentes a la vez. Por ejemplo, los sistemas operativos multitarea típicos pueden ejecutar varios programas de aplicación a la vez en una sola máquina. En vista de esto y de la evolución de una red abierta compartida (es decir, Internet), la seguridad y la privacidad se han convertido en dos cuestiones importantes y difíciles que afronta la industria de los ordenadores. A medida que el ordenador personal cobra importancia central en el hogar, el trabajo y la enseñanza, los consumidores y los clientes en el mundo empresarial son cada vez más conscientes, por igual, de los problemas de privacidad y seguridad. Mejorar la capacidad del soporte lógico y del soporte físico de proteger la integridad de la información digital y la privacidad de los usuarios de ordenadores se ha convertido en una prioridad vital tanto de los desarrolladores de soportes lógicos como de los fabricantes de soportes físicos. La Microsoft Corporation, de Redmond, Washington, ha introducido la plataforma de ordenadores personales Base Informática Segura de Próxima Generación (NGSCB) que proporciona seguridad y privacidad en un sistema operativo.

- 20 Según se muestra en la Figura 2, en la NGSCB convencional dentro de un ordenador 110 un sistema de seguridad de la "parte derecha" (PD) trabaja conjuntamente con un sistema tradicional de la "parte izquierda" (PI) y una unidad central de proceso (CPU). La PD está diseñada para proteger contra un soporte lógico malicioso mientras se mantiene la apertura del sistema operativo. Con la NGSCB, las aplicaciones se ejecutan en un espacio protegido de memoria que es sumamente resistente a la alteración de un soporte lógico y a la interferencia. Típicamente, hay un conjunto de chips en el ordenador 110 que usan tanto la PI como la PD. La PI y la PD son una división o partición lógica, aunque impuesta físicamente, del ordenador 110.

La PI comprende aplicaciones tradicionales 205, 210, tales como Microsoft® Word® y Microsoft® Excel®, junto con un sistema operativo convencional 201, tal como el sistema operativo Microsoft® Windows®. Aunque se muestran dos aplicaciones, típicamente puede implementarse un número cualquiera.

- 30 La PD comprende agentes 255, 260 de confianza, junto con un "nexo" 251. Un nexo es un sistema operativo de "alta seguridad" que proporciona cierto nivel de seguridad en cuanto a su comportamiento y puede comprender todo el código en modo de núcleo en la PD. Por ejemplo, podría emplearse un nexo para trabajar con información secreta (por ejemplo, claves criptográficas, etc.) que no debiera divulgarse, proporcionando una memoria cubierta que está garantizado que no filtre información fuera del nexo y permitiendo que solo ciertas aplicaciones certificadas se ejecuten bajo el nexo y accedan a la memoria cubierta. El nexo 251 no debería interactuar con el sistema operativo principal 201 de ninguna manera que permitiera que ocurriesen eventos en el sistema operativo principal 201 que comprometiesen el comportamiento del nexo 251. El nexo 251 puede permitir que se ejecuten todas las aplicaciones, o el propietario de una máquina puede configurar una directiva de máquina en la que el nexo 251 permite que se ejecuten únicamente ciertos agentes. En otras palabras, el nexo 251 ejecutará cualquier agente que el propietario de la máquina le diga que ejecute. El propietario de la máquina puede también decir al nexo qué no ejecutar.

- 45 El nexo 251 aísla los agentes 255, 260 de confianza, gestiona las comunicaciones hacia los agentes 255, 260 de confianza y procedentes de los mismos, y cifra criptográficamente los datos almacenados (por ejemplo, almacenados en una unidad de disco duro). Más en particular, el nexo 251 se ejecuta en modo de núcleo en un espacio de confianza y proporciona servicios básicos a los agentes 255, 260 de confianza, como el establecimiento de los mecanismo de proceso para comunicarse con agentes de confianza y otras aplicaciones, y servicios especiales de confianza, como la ratificación de una plataforma de soporte físico / soporte lógico o un entorno de ejecución y el descifrado de secretos. La ratificación es la capacidad de una porción de código para firmar o ratificar de otra forma un dato y para garantizar, además, al receptor que los datos fueron construidos por medio de una pila de soporte lógico infalsificable identificada de manera criptográfica.

- 50 Un agente de confianza es un programa, o parte de un programa, o un servicio que se ejecuta en modo de usuario en un espacio de confianza. Un agente 255, 260 de confianza llama al nexo 251 para servicios relacionados con la seguridad y servicios críticos generales, como la gestión de la memoria. Un agente de confianza es capaz de almacenar secretos usando un almacenamiento cifrado y se autentica a sí mismo usando los servicios de ratificación del nexo. Cada agente o entidad de confianza controla su propio dominio de confianza y no es preciso que tengan dependencias mutuas.

La PD comprende, además, un componente 253 de soporte de seguridad (SSC) que usa un par de claves de infraestructura de clave pública (PKI) junto con funciones de codificación para proporcionar un estado seguro.

La NGSCB proporciona características como "ratificación", "almacenamiento cifrado" y "aislamiento fuerte de procesos". La ratificación permite que otros ordenadores sepan que un ordenador es realmente el ordenador que pretende ser y está ejecutando el soporte lógico que pretende estar ejecutando. Dado que el soporte lógico y el soporte físico de la NGSCB son criptográficamente verificables para el usuario y los demás ordenadores, programas y servicios, el sistema puede verificar que otros ordenadores y procesos son fiables antes de acoplarse a los mismos y compartir información. Así, la ratificación permite que el usuario revele características seleccionadas del entorno operativo a peticionarios externos.

El almacenamiento cifrado permite que el usuario codifique información para que solo pueda ser objeto de acceso por parte de una aplicación fiable. Esto puede incluir solo la aplicación que creó la información en su origen o cualquier aplicación en la que confíe la aplicación propietaria de los datos. Por lo tanto, el almacenamiento cifrado permite que un programa almacene secretos que no puedan ser recuperados por programas en los que no se confía, como un virus o un troyano.

El aislamiento fuerte de procesos proporciona un espacio de confianza para forjar un área segura (la PD). Las operaciones que se ejecutan en la PD están protegidas y aisladas de la PI, lo que las hace significativamente más seguras ante un ataque.

La NGSCB también proporciona una entrada y una salida seguras. Con la NGSCB, las pulsaciones de tecla son cifradas antes de que puedan ser leídas por el soporte lógico y descifradas una vez que alcanzan la PD. Esto significa que no puede usarse un soporte lógico malicioso para grabar, robar o modificar pulsaciones de tecla. La salida segura es similar. La información que aparece en pantalla puede ser presentada al usuario para que nadie más pueda interceptarla ni leerla. Tomadas en conjunto, estas cosas permiten que un usuario sepa con un alto grado de confianza que el soporte lógico de su ordenador está haciendo lo que se supone que hace.

A pesar de los recursos de confianza sustancial disponibles a la PD, sigue sin confiarse en la PI. La presente invención aborda esta y otras deficiencias de los actuales sistemas informáticos fiables.

El documento US 6 308 208 B1 versa acerca de un procedimiento para monitorizar recursos informáticos distribuidos por la red usando agentes celulares distribuidos. Se asocia un conjunto de una o más "células" con un conjunto de recursos informáticos dados que comprende un recurso maestro. Preferentemente, cada célula está asociada con un recurso respecto del conjunto de recursos informáticos dados. Se asocia una célula de mayor nivel correspondiente al recurso maestro con una o más de las células de menor nivel. Un agente distribuido particular no monitorización no sabe necesariamente a qué otras células está afectando. Un mecanismo de vigilancia de células permite que una consola de operador monitorice el sistema automático celular distribuido, haciéndose preferentemente que la consola de operador aparezca al sistema automático como células ordinarias de vigilancia de células implicadas en la monitorización de recursos. Los cambios a esas células se propagan entonces a la consola de operador, que preferentemente usa técnicas de interfaz visual de usuario para proporcionar a un operador una visión dinámica del estado de los recursos seleccionados de la red.

Resumen de la invención

Es el objeto de la presente invención proporcionar un mecanismo para proyectar la fiabilidad de entidades en un entorno de confianza a entidades en un entorno sin confianza.

Este objeto se resuelve mediante la materia de las reivindicaciones independientes.

Las realizaciones se dan en las reivindicaciones dependientes.

Se describen sistemas y procedimientos en los que se proporcionan un entorno sin confianza y un entorno con confianza. En el entorno con confianza se ejecuta un agente base de monitorización. El agente base de monitorización monitoriza el entorno sin confianza.

Según una realización, el agente de monitorización está asociado con una aplicación y el agente de monitorización monitoriza su aplicación asociada en busca de eventos o comportamientos que indiquen un ataque. La naturaleza de confianza del agente de monitorización permite que estos eventos/comportamientos sean detectados y sean objeto de informe de manera fiable, proyectándose por ello la fiabilidad del entorno de confianza al entorno sin confianza. El agente base de monitorización puede autorizar, prohibir o modificar un evento del entorno sin confianza objeto de informe al agente de monitorización o descubierto por este. "Objeto de informe a" cubre los casos tales como que el soporte físico informe de un tentativa de mover la GDT (tabla de descriptores globales) al nexo, el cual, a su vez, la haría objeto de informe, por ejemplo, al agente base de monitorización. El descubrimiento sería un caso en el que el agente base de monitorización (para el SO) o el agente de monitorización (para alguna aplicación) descubre un problema barriendo, por ejemplo, la memoria de la aplicación sin confianza.

Para otra realización, el agente base de monitorización responde a la entrada recibida de la entrada segura. Por ejemplo, el agente base de monitorización puede negarse a permitir cambios del entorno sin confianza sin recibir autorización por medio de una entrada segura. Como ejemplo adicional, el agente base de monitorización puede negarse a permitir cambios del entorno sin confianza a no ser que los cambios estén descritos por un paquete que esté firmado por una entidad autorizada.

Para otra realización adicional, el agente de monitorización usa un almacenamiento cifrado para guardar un secreto para un sistema operativo o una aplicación que reside en el entorno sin confianza. El agente de monitorización puede negarse a revelar el secreto al sistema operativo o a la aplicación, a no ser que el sistema operativo o la aplicación tengan un compendio que coincida con el propietario del secreto. De manera alternativa, el agente de monitorización puede negarse a revelar el secreto al sistema operativo o a la aplicación, a no ser que el sistema operativo o la aplicación estén en una lista de compendios que pueden leer el secreto.

Según otras características, el agente de monitorización usa una prueba para determinar si una entidad legítima está solicitando el secreto. Una prueba tal incluye el examen de las pilas de la entidad y garantizar que las pilas tengan contenidos legales de pila. Además, el agente de monitorización puede editar un estado del entorno sin confianza para hacerlo seguro o aceptable de otra manera. El estado puede comprender una configuración inicial o una opción de informe de error.

A partir de la siguiente descripción detallada de realizaciones ilustrativas que prosigue con referencia a los dibujos adjuntos se harán evidentes características y ventajas adicionales de la invención.

Breve descripción de los dibujos

El anterior resumen, así como la siguiente descripción detallada de las realizaciones preferentes, se entiende mejor cuando se lee en conjunto con los dibujos adjuntos. Con el fin de ilustrar la invención, en los dibujos se muestran construcciones ejemplares de la invención; sin embargo, la invención no está limitada a los procedimientos e instrumentalidades específicos dados a conocer. En los dibujos:

la Figura 1 es un diagrama de bloques que muestra un entorno informático ejemplar en el cual pueden implementarse aspectos de la invención;

la Figura 2 es un diagrama de bloques de un sistema existente de NGSCB que tiene entornos tanto con confianza como sin confianza;

la Figura 3 es un diagrama de bloques de un sistema existente ejemplar de proyección según la presente invención; y

la Figura 4 es un diagrama de flujo de un procedimiento ejemplar de proyección según la presente invención.

Descripción detallada de realizaciones preferentes

Visión general

En una sola máquina que tiene entidades ejecutándose en un entorno sin confianza y entidades que se ejecutan en un entorno de confianza, la presente invención proporciona un mecanismo para proyectar la fiabilidad de las entidades en el entorno de confianza a las entidades en el entorno sin confianza. La invención está dirigida a mecanismos usados cuando un primer entorno de ejecución (por ejemplo, un sistema operativo) hospeda un segundo entorno de ejecución. La invención se aplica a situaciones como la Base Informática Segura de Próxima Generación (NGSCB), de Microsoft®, en la que un sistema operativo regular (por ejemplo, el sistema operativo Windows®) hospeda un sistema operativo seguro (por ejemplo, el nexos). Se describen diversos mecanismos que permiten que el segundo entorno proyecte su fiabilidad al segundo entorno.

Entorno informático ejemplar

La Figura 1 es un ejemplo de un entorno adecuado 100 de un sistema informático en el cual pueden implementarse aspectos de la invención. El entorno 100 del sistema informático es solo un ejemplo de un entorno informático adecuado y no se pretende sugerir ninguna limitación en cuanto al alcance del uso o la funcionalidad de la invención. Tampoco debiera interpretarse que el entorno informático 100 tenga ninguna dependencia ni requisito relativos a un componente cualquiera o a una combinación de componentes ilustrados en el entorno operativo ejemplar 100.

La invención es operativa con numerosos entornos o configuraciones adicionales de uso general o de uso especial. Ejemplos de sistemas, entornos y/o configuraciones informáticos bien conocidos que pueden ser adecuados para su uso con la invención incluyen, sin limitación, ordenadores personales, ordenadores servidores, dispositivos de mano o portátiles, sistemas multiprocesador, sistemas basados en microprocesadores, decodificadores, teléfonos móviles,

componentes electrónicos programables de consumo, PC de red, miniordenadores, ordenadores centrales, entornos informáticos distribuidos que incluyan cualquiera de los sistemas o dispositivos anteriores, y similares.

La invención puede ser descrita en el contexto general de instrucciones ejecutables por ordenador, como módulos de programa, que son ejecutadas por un ordenador. Generalmente, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que llevan a cabo tareas particulares o implementan tipos particulares de datos abstractos. La invención también puede ser puesta en práctica en entornos informáticos distribuidos en los que las tareas son realizadas por dispositivos remotos de proceso que están unidos por medio de una red de comunicaciones u otro medio de transmisión de datos. En un entorno informático distribuido, los módulos de programa y otros datos pueden estar situados en medios de almacenamiento de ordenador tanto locales como remotos, incluyendo dispositivos de almacenamiento en memoria.

Con referencia a la Figura 1, un sistema ejemplar para implementar la invención incluye un dispositivo informático de uso general en la forma de un ordenador 110. Los componentes del ordenador 110 pueden incluir, sin limitación, una unidad 120 de proceso, una memoria 130 del sistema y un bus 121 del sistema que acopla diversos componentes del sistema, incluyendo la memoria del sistema, a la unidad 120 de proceso. El bus 121 del sistema puede ser de cualquiera de varios tipos de estructuras de bus, incluyendo un bus de memoria o un controlador de memoria, un bus de periféricos y un bus local usando cualquiera de una variedad de arquitecturas de bus. A título de ejemplo, y no de limitación, tales estructuras incluyen el bus de Arquitectura Industrial Normalizada (ISA), el bus de Arquitectura de Microcanal (MCA), el bus de ISA Mejorada (EISA), el bus local de la Asociación de Normativa Electrónica sobre Vídeo (VESA) y el bus de Interconexión de Componentes Periféricos (PCI) (también conocido como bus de entresuelo).

Típicamente, el ordenador 110 incluye varios medios legibles por ordenador. Los medios legibles por ordenador pueden ser cualesquiera medios disponibles que puedan ser objeto de acceso por parte de un ordenador 110 e incluyen medios tanto volátiles como no volátiles, medios extraíbles y no extraíbles. A título de ejemplo, y no de limitación, los medios legibles por ordenador pueden comprender medios de almacenamiento de ordenador y medios de comunicaciones. Los medios de almacenamiento de ordenador incluyen medios tanto volátiles como no volátiles, extraíbles y no extraíbles implementados en cualquier procedimiento o tecnología para el almacenamiento de información, como instrucciones, estructuras de datos, módulos de programa legibles por ordenador u otros datos. Los medios de almacenamiento de ordenador incluyen, sin limitación, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM, discos versátiles digitales (DVD) o almacenamiento en discos ópticos, casetes magnéticos, cinta magnética, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar la información deseada y al que pueda acceder el ordenador 110. Típicamente, los medios de comunicación implementan instrucciones, estructuras de datos, módulos de programa legibles por ordenador u otros datos en una señal modulada de datos, como una onda portadora u otro mecanismo de transporte e incluyen cualesquiera medios de distribución de la información. La expresión "señal modulada de datos" significa una señal una o más de cuyas características ha sido configurada o modificada de tal manera que codifica la información de la señal. A título de ejemplo, y no de limitación, los medios de comunicaciones incluyen medios alámbricos, como una red cableada o una conexión cableada directa, y medios inalámbricos, como medios acústicos, de RF, infrarrojos y otros medios inalámbricos. Las combinaciones de cualquiera de los anteriores también debería incluirse dentro del alcance de los medios legibles por ordenador.

La memoria 130 del sistema incluye medios de almacenamiento de ordenador en forma de memoria volátil y/o no volátil, como la ROM 131 y la RAM 132. Típicamente, en la ROM 131 se almacena un sistema básico 133 de entrada/salida (BIOS), que contiene las rutinas básicas que contribuyen a transferir información entre elementos dentro del ordenador 110, como durante el arranque. Típicamente, la RAM 132 contiene datos y/o módulos de programa que son inmediatamente accesibles a la unidad 120 de proceso y/o están siendo operados en el momento por la misma. A título de ejemplo, y no de limitación, la Figura 1 ilustra el sistema operativo 134, programas 135 de aplicación, otros módulos 136 de programa y datos 137 de programa.

El ordenador 110 también puede incluir otros medios de almacenamiento de ordenador extraíbles/no extraíbles volátiles/no volátiles. A título de ejemplo únicamente, la Figura 1 ilustra una unidad 140 de disco duro que lee de medios no extraíbles no volátiles o escribe en los mismos, una unidad 151 de disco magnético que lee de un disco magnético extraíble no volátil 152 o escribe en el mismo, y una unidad 155 de disco óptico que lee de un disco óptico extraíble no volátil 156, como un CD-ROM u otros medios ópticos, o escribe en el mismo. Otros medios de almacenamiento de ordenador extraíbles/no extraíbles volátiles/no volátiles que pueden ser usados en el entorno operativo ejemplar incluyen, sin limitación, casetes de cinta magnética, tarjetas de memoria flash, discos versátiles digitales, cinta de vídeo digital, RAM de estado sólido, ROM de estado sólido y similares. La unidad 141 de disco duro está conectada típicamente al bus 121 del sistema a través de una interfaz de memoria no extraíble, como la interfaz 140, y la unidad 151 de disco magnético y la unidad 155 de disco óptico están típicamente conectadas al bus 121 del sistema por medio de una interfaz de memoria extraíble, como la interfaz 150. Se contempla, además, que la presente invención pueda implementarse también en un procesador integrado en el cual la CPU y toda la memoria estén en una sola matriz en un solo paquete.

Las unidades y sus medios de almacenamiento de ordenador asociados, presentados más arriba e ilustrados en la Figura 1, proporcionan almacenamiento de instrucciones, estructuras de datos, módulos de programa legibles por ordenador u otros datos para el ordenador 110. En la Figura 1, por ejemplo, la unidad 141 de disco duro está ilustrada almacenando el sistema operativo 144, programas 145 de aplicación, otros módulos 146 de programa y datos 147 de programa. Obsérvese que estos componentes pueden ser iguales o diferentes que el sistema operativo 134, los programas 135 de aplicación, otros módulos 136 de programa y los datos 137 de programa. El sistema operativo 144, los programas 145 de aplicación, otros módulos 146 de programa y los datos 147 de programa reciben aquí números diferentes para ilustrar que, como mínimo, son copias diferentes. Un usuario puede introducir órdenes e información en el ordenador 110 a través de dispositivos de entrada, como un teclado 162 y un dispositivo 161 de puntero, denominado comúnmente ratón, bola de mando o panel táctil. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, una palanca de mando, un mando para juegos, una antena parabólica, un escáner o similares. Estos y otros dispositivos de entrada están conectados a menudo a la unidad 120 de proceso a través de una interfaz 160 de entrada de usuario que está acoplada al bus del sistema, pero pueden estar conectados por medio de otra interfaz y otras estructuras de bus, como un puerto paralelo, un puerto de juegos o un bus serie universal (USB). Un monitor 191 u otro tipo de dispositivo de visualización también están conectados al bus 121 del sistema por medio de una interfaz, como una interfaz 190 de vídeo. Además del monitor, los ordenadores pueden también incluir otros dispositivos periféricos de salida, como altavoces 197 y una impresora 196, que pueden estar conectados a través de una interfaz 195 para periféricos de salida.

El ordenador 110 puede operar en un entorno de red usando conexiones con uno o más ordenadores remotos, como un ordenador remoto 180. El ordenador remoto 180 puede ser un ordenador personal, un servidor, un dispositivo de encaminamiento, un PC de red, un dispositivo del mismo nivel u otro nodo común de red, y típicamente incluye muchos o la totalidad de los elementos descritos más arriba en relación con el ordenador 110, aunque en la Figura 1 solo se ha ilustrado un dispositivo 181 de almacenamiento de memoria. Las conexiones lógicas representadas incluyen una red 171 de área local (LAN) y una red 173 de área amplia (WAN), pero también pueden incluir otras redes. Tales entornos de red son comunes en oficinas, redes de ordenadores de ámbito empresarial, intranets e Internet.

Cuando se usa en un entorno de red LAN, el ordenador 110 está conectado a la LAN 171 a través de una interfaz o un adaptador 170 de red. Cuando se usa en un entorno de red WAN, el ordenador 110 incluye típicamente un módem 172 u otros medios para establecer comunicaciones por la WAN 173, como Internet. El módem 172 puede ser interno o externo, puede estar conectado al bus 121 del sistema a través de la interfaz 160 de entrada de usuario u otro mecanismo apropiado. En un entorno de red, los módulos de programa representados en relación con el ordenador 110 o porciones del mismo pueden ser almacenados en el dispositivo remoto de almacenamiento de memoria. A título de ejemplo, y no de limitación, la Figura 1 ilustra programas 185 de aplicación que residen en el dispositivo 181 de memoria. Se apreciará que las conexiones de red mostradas son ejemplares y que pueden usarse otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.

Realizaciones ejemplares

Según se ha descrito previamente, se conoce en la técnica que un ordenador puede estar configurado para proporcionar dos entornos diferenciados: con confianza y sin confianza. El código ordinario cuya fiabilidad no ha sido verificada (es decir, código cuyo comportamiento no ha sido verificado, o del que no puede descartarse que posiblemente sirva un fin malévolo) se ejecuta en el entorno sin confianza. El soporte lógico ordinario de aplicación, como juegos, tratamientos de texto, hojas de cálculo, etc., así como los sistemas operativos ordinarios, los controladores de dispositivos y los depuradores están incluidos por lo general en el entorno sin confianza. El código cuya fiabilidad ha sido verificada de alguna manera puede ejecutarse en el entorno de confianza. Alguna porción de la memoria del ordenador (es decir, la memoria "aislada" o "cubierta") está diseñada para que sea accesible solamente al entorno de confianza.

Para la siguiente exposición, un agente es "de confianza" si ha sido instanciado según un procedimiento seguro diseñado para conservar su integridad o hacer evidente cualquier violación de su integridad. Por ejemplo, el agente puede ser iniciado a través de un procedimiento de confianza que verifique la identidad del agente y el entorno en el que se está ejecutando (ratificación), puede asignársele una ubicación segura de memoria (memoria cubierta) que no sea accesible a ningún otro agente, de confianza o carente de ella, y puede ser capaz de cifrar secretos. Un agente de confianza de ese tipo puede ser identificado de forma única y fiable.

En el entorno de confianza hay limitaciones a lo que se le permite hacer al código. Por ejemplo, hay menos API de confianza (con respecto a riquísimo conjunto de las API en una PI típica), los agentes que se ejecutan en el entorno de confianza solo pueden comunicarse entre sí a través de mecanismos formales restringidos de Comunicaciones entre Procesos (IPC) y los agentes pueden tener acceso a un conjunto más restringido y primitivo de API y servicios para presentar texto e imágenes al usuario. Estas limitaciones reducen la complejidad y, en consecuencia, la superficie de ataque del entorno de confianza y de los agentes de confianza que operan dentro de él. Por otro lado, el entorno sin confianza es similar al entorno creado típicamente por el sistema operativo en un sistema informático "abierto" (por ejemplo, un ordenador personal, un ordenador de mano, etc.); es decir, se permite que se ejecute casi cualquiera código en tal entorno sin confianza, y el código que se ejecuta en el entorno estándar tiene pleno acceso

a un conjunto grande y rico de servicios e interfaces de programación. El entorno sin confianza y el entorno de confianza pueden ser divididos adicionalmente en subentornos. Por ejemplo, el entorno sin confianza puede ser dividido en un modo de usuario sin confianza (en el que se ejecutan las aplicaciones ordinarias) y en un modo de núcleo sin confianza (en el que se ejecuta el sistema operativo ordinario). De manera similar, el entorno de confianza puede ser dividido en un modo de usuario con confianza (en el que se ejecutan aplicaciones especiales de confianza) y en un modo de núcleo con confianza (en el que se ejecuta el sistema operativo que crea el entorno de confianza para las aplicaciones de confianza).

Cuando en el mismo sistema de ordenador coexisten entornos de confianza y sin confianza, el entorno de confianza puede adoptar medidas para garantizar que su fiabilidad no pueda verse afectada por nada que ocurra en el entorno sin confianza ni por cualquier código del modo de usuario en el entorno de confianza. Las realizaciones de la presente invención proporcionan un mecanismo para proyectar o usar de otro modo la fiabilidad del lado de confianza en beneficio del lado sin confianza.

La Figura 3 es un diagrama de bloques de una realización de un sistema de proyección según la presente invención y la Figura 4 es un diagrama de flujo de una realización del procedimiento de proyección según la presente invención. La PI del sistema que se ejecuta en el ordenador 110 es similar a la descrita más arriba con respecto a la Figura 2. Dos aplicaciones 305, 310 se están ejecutando en conjunto con un sistema operativo 301. Las porciones de la PD son también similares a las descritas con respecto a la Figura 2. Dos agentes 355, 360 de confianza se están ejecutando junto con un nexo 351 y un SSC 353. Se contempla que en la PI pueda ejecutarse un número cualquiera de aplicaciones y que en la PD pueda ejecutarse un número cualquiera de agentes de confianza.

La Figura 3 muestra un sistema en el que el sistema operativo 301 y el nexo 351 se ejecutan en un solo ordenador 110. Una separación lógica 350 entre el sistema operativo 301 y el nexo 351 permite que ocurran ciertas comunicaciones mientras se protege al nexo 351 de eventos que se originan en el sistema operativo 301.

En la realización ilustrada por la Figura 3, el sistema operativo 301 es un sistema operativo anfitrión y el nexo 351 es un huésped hospedado por el SO 301. Es decir, el SO 301 proporciona ciertos servicios y recursos del nexo 351, como la memoria y el tiempo de procesador. Para una realización, la separación lógica 350 permite que el nexo 351 cuenta con ciertos recursos del sistema operativo 301, mientras sigue permitiendo que el nexo 351 se proteja de acciones (ya sean maliciosas o inocentes) que surgen en el sistema operativo 301 y que podrían hacer que el nexo 351 se comporte de manera contraria a sus especificaciones conductuales. Por ejemplo, el nexo 351 y los recursos de confianza asociados con él, por ejemplo el SSC 353, pueden gestionar la separación lógica. Sin embargo, se entenderá que la invención no está limitada a la forma particular del nexo 351. Se contemplan mecanismos que permiten que se construya la separación 350 para permitir este equilibrio de interacción y protección.

Debería notarse que la Figura 3 muestra al sistema operativo 301 como un "anfitrión" y al nexo 351 como un "huésped". En general, esta caracterización se refiere al hecho de que, en estos ejemplos, el sistema operativo 301 proporciona cierta infraestructura de sistema operativo que es usada por los sistemas operativos 301 y el nexo 351 (por ejemplo, controladores de dispositivos, programación, etc.). El nexo 351 es un "huésped" en el sentido de que puede contar con ciertos recursos de infraestructura del sistema operativo 301 en vez de proporcionarlos él mismo. Sin embargo, debería hacerse notar que los parámetros de lo que convierte a un sistema operativo en un "anfitrión" o un "huésped" son flexibles. Debería apreciarse que las técnicas descritas en el presente documento pueden ser aplicadas a la interacción de cualesquiera dos o más sistemas operativos que se ejecutan en la misma máquina (o incluso en el mismo conjunto de máquinas conectadas). Dos o más sistemas operativos que se ejecutan en una sola máquina son ejemplos de "entornos" que pueden precisar interactuar entre sí en una sola máquina, aunque se entenderá que la invención no está limitada a los sistemas operativos tradicionales.

La proyección es el mecanismo mediante el cual algunos de los poderes y las propiedades de los agentes de confianza (en la PD) pueden extenderse al código de la PI. Según un ejemplo, la proyección permite que los poderes de la plataforma de ordenadores personales NGSCB sean aplicados al código existente. Por ejemplo, en vez de portar una aplicación como Microsoft® Excel® a la PD, la proyección según la presente invención permite la construcción de un agente de monitorización (también denominada ángel en el presente documento) para la aplicación (también denominada mortal en el presente documento), el cual permite, a su vez, que la aplicación existente se ejecute con muchas de las mismas propiedades útiles que un agente de confianza. La proyección puede aplicarse tanto al sistema operativo de la PI (por ejemplo, Microsoft® Windows®) como a los programas de aplicación de la PI (por ejemplo, Microsoft® Office®) para los que es deseable cierto nivel de operación de confianza. La proyección también puede ser aplicada a los controladores de dispositivos de la PI. Así, como se describe de forma adicional más abajo, la proyección permite que los agentes de confianza proyecten, garanticen, ratifiquen y extiendan los sistemas operativos, los servicios y los programas de la PI.

La Figura 3 muestra un agente 390 de monitorización que corresponde a la aplicación 305 y un agente 395 de monitorización que corresponde a la aplicación 310 (etapa 400 en la Figura 4). Cada agente o ángel de monitorización protege a su aplicación asociada.

Para una realización, el creador de la entidad de interés de la PI (por ejemplo, una aplicación) también crea un ángel que guarda a la entidad de la PI. Esto permite que el creador proporcione el ángel con un profundo conocimiento de

la aplicación que monitoriza. Tal ángel puede ser más sensible a anomalías en la aplicación que monitoriza y así protegerla y validarla de forma más efectiva. Por ejemplo, un agente base de monitorización creado por un desarrollador del sistema operativo puede incorporar un conocimiento detallado sobre la gestión de la memoria del sistema operativo que le permite identificar rápidamente operaciones sospechosas en la memoria.

5 Para otra realización, un ángel puede tomar acción correctiva y preventiva si detecta una actividad anómala o sospechosa en su aplicación asociada. Por ejemplo, el ángel puede detectar una tentativa de su aplicación asociada por alterar una variable clave en la memoria que sea considerada invariante por el creador de la aplicación, e interceptar la escritura en tal variable. Tal operación de escritura probablemente indicaría, al menos, corrupción del código de la aplicación, si no una subversión descarada, por parte de código malévolo, por ejemplo vírico. En
10 resumidas cuentas, un ángel actúa como un agente de monitorización para vigilar actividades negativas o sospechosas en su aplicación asociada y tomar la debida acción correctiva o preventiva. Sus acciones pueden ser circunscritas para evitar que el ángel dañe a su aplicación asociada. Un ángel puede estar unido, por ejemplo, a una entidad, un programa o una aplicación particulares, o a un grupo de tales entidades, programas y/o aplicaciones.

15 Un agente base 380 de monitorización (también denominado arcángel en el presente documento) está asociado con el sistema operativo base (es decir, el SO 301 de la PI) (bloque 410). Para una realización, el agente base 380 de monitorización está escrito por el creador del sistema operativo de la PI. Esto permite que el agente base 380 de monitorización incorpore un conocimiento detallado sobre el sistema operativo de la PI, lo que lo hace más sensible a un comportamiento anómalo del sistema operativo asociado.

20 Por ejemplo, un arcángel podría conocer el formato de la base de datos de direcciones virtuales, la base de datos de procesos y la base de datos de PFN (números de tramas de página) y, en base a esto, detectar casos en los que controladores maliciosos de dispositivos hubiesen establecido correspondencias ilegales con los procesos dando a los PFN correspondencias que no deberían tener. Así, el arcángel podría detectar correspondencias no realizadas por el gestor de memoria (por ejemplo, por el controlador malicioso de dispositivos) y podría detectar correspondencias cruzadas de procesos que no deberían existir.

25 En tal caso, un arcángel podría conspirar con un SO moral alterado. El SO y el arcángel podrían acordar, por ejemplo, que la base de datos de PFN sea siempre coherente en todo momento en que no se mantenga un bloqueo particular, y que esta coherencia sea representable mediante una suma de control. Así, a intervalos periódicos, el arcángel podría inspeccionar el bloqueo y, al encontrarlo desbloqueado (es una variable de memoria y, por ello, fácil de comprobar) ir a la base de datos de PFN y realizar en ella una suma de control. Si el arcángel descubre que la
30 suma de control no coincide, sabe que la base de datos de PFN ha sido alterada.

Además, un arcángel podría conocer las variables de control para el depurador del núcleo y obligar a las variables de control a inhabilitar el uso del depurador del núcleo.

35 Un ejemplo adicional incluye la carga de procesos: monitorizar el cargador, el gestor de la memoria intermedia, el gestor de errores de paginación, etc., para garantizar que en un proceso del modo de usuario (o en cualquier otro módulo cargado en el sistema) se carguen correctamente los bits correctos o debidamente firmados, quizá enumerados en una lista de claves calculadas guardadas en una tabla conocida por el arcángel. El arcángel sería capaz de prever cuándo el cargador, el gestor de errores de paginación, etc., precisarían establecer una correspondencia código/datos en un proceso o fuera de él (paginación, etc.). La PD podría mantener bloqueadas las páginas físicas de la PI para ese proceso (incluso para el SO de la PI) a no ser que el SO estuviera realizando
40 funciones buenas conocidas. La PD controla las tablas de páginas para los procesos de la PI. Así, hay varios mecanismos que un escritor de arcángeles podría incorporar al arcángel para restringir un mal comportamiento.

45 Un ejemplo adicional incluye el endurecimiento de procesos. Hay mecanismos conocidos y aprobados para un proceso que modifica otro proceso. El arcángel puede garantizar que estén restringidos todos los mapas de memoria compartida, así como la copia de datos a un espacio de proceso diferente o desde el mismo. Otro ejemplo implica el núcleo de solo lectura, en el cual todas las páginas de "texto" (páginas de código) del núcleo y de los controladores de dispositivos están bloqueadas.

50 El arcángel 380 también soporta la proyección a los ángeles por proceso (acceso restringido). Esto significa que los ángeles, que son como agentes porque el sistema ejecutará cualquier ángel que el usuario le pida (en coherencia con la directiva de usuarios) y que no forme parte del vector de ratificación, según se define más abajo (es decir, el arcángel es, en efecto, parte de la configuración de la máquina), podrían sembrar el caos, invadiendo la privacidad de la parte izquierda, husmeando, por ejemplo, en las aplicaciones mortales en las que se supone que no se aplican. Por lo tanto, es deseable que los ángeles estén muy ligados a aplicaciones (mortales) particulares. Esto se hace, preferentemente, permitiendo que un ángel solo afecte a un mortal que arranque el ángel o permitiendo que un ángel se aplique únicamente a un mortal que coincida con un compendio declarado en el manifiesto del ángel,
55 realizando el arcángel la comprobación del compendio, y solo después de las llamadas a la aplicación mortal con el compendio del ángel para arrancarlo. Esto es deseable porque hace seguro y práctico para permitir a cualquier creador de aplicaciones escribir un ángel para su aplicación y permitir que cualquier usuario lo use sin arriesgar crear el caos o destruir la privacidad de todo lo demás.

Así, el arcángel es tanto el agente que vigila la PI como un agente que ofrece servicios a otros ángeles. Dado que el arcángel tiene un conocimiento sumamente detallado de las estructuras de los procesos de la PI, probablemente sea el arcángel quien decide qué ángel puede ligarse a qué proceso de la PI. La restricción significa que un ángel (que no es parte del vector de ratificación del nexa) solo puede tocar los procesos que arranque o que lo invoquen para que los proteja. Esto impide que los ángeles actúen de forma aleatoria en los procesos de la PI. Esta división (el arcángel obtiene poderes de nivel de SO y es validado como el nexa; los ángeles obtienen poderes restringidos de nivel de aplicación y pueden ejecutarse libremente como cualquier otro agente) es deseable.

Para una realización, los ángeles pueden incluir al arcángel (y, por extensión, el SO base de la PI) en sus vectores de ratificación. Un vector de ratificación es una lista de compendios de componentes relevantes para la seguridad que establecen la configuración relevante de seguridad de una entidad. Por ejemplo, el compendio para un agente podría incluir la máquina o la propia placa base, el nexa, y el propio agente, junto con otra información. Esta pila de números es un indicador fuerte y fiable de qué agente es y en qué entorno se está ejecutando el agente. Permite que otra entidad confíe o no en que está tratando con "el agente real". Los vectores de ratificación se apilan (de modo que el compendio del agente no es parte del vector para el nexa, sino que el compendio del nexa es parte del compendio para el agente). Por lo tanto, cuando el vector de ratificación de algo está incluido en otra cosa, esto significa que todos están ligados entre sí en una configuración reconocible de seguridad. Una propiedad de una ratificación es que identifica con mucha certeza la configuración relevante de seguridad de un sistema.

Dicho de otra forma, un vector de ratificación es una lista de valores de compendio que definen una identidad del soporte lógico de la PD. Preferentemente, el soporte lógico cargado en la PD es compendiado antes de ser cargado y el propio proceso está bien aislado para que no pueda cambiar. Este es un proceso inductivo: el soporte físico firma el compendio del nexa (ratifica el compendio del nexa) y el nexa, a su vez, ratifica al agente. De esta manera, una entidad externa puede validar estos compendios contra una lista conocida para determinar si esa entidad externa autoriza el soporte lógico que se ejecuta en el sistema. El ángel y el arcángel, puesto que se ejecutan en la PD, tienen identidades de código bien definidas. Por esta razón, estas identidades de código pueden ser enumeradas en el vector de ratificación que describe el entorno en el que se ejecuta el código de la PI. Dado que el ángel no puede controlar completamente la ejecución del código de la PI, esta declaración de identidad de código no es tan fuerte como una declaración de identidad de código de un agente de la PD, pero no significa que el código dado de la PI se esté ejecutando bajo las restricciones del ángel, el arcángel y el nexa que no tienen identidades fuertes de código.

Las realizaciones de un arcángel pueden exponer algunos conjuntos de las API a los ángeles para proporcionar soporte a algunas funciones y/o características de los ángeles. Por ejemplo, para cualquier operación de memoria, el arcángel mediará, de forma deseable. Un ángel podría desear examinar el código de la aplicación cubierta en la dirección virtual VA=100. Sin embargo, puede no conocerse con qué relación física se corresponde la misma. El nexa no sabe de tales estructuras. Por lo tanto, en vez de ello, el arcángel (que conoce cómo opera el SO de la PI) usa servicios básicos del nexa (que solo pueden invocar los arcángeles) para leer la memoria relevante del núcleo de la PI. El arcángel usa datos de la memoria del SO de la PI para calcular las correspondencias correctas para la memoria de aplicaciones de la PI. Al ángel se le dice entonces qué dirección de aplicación cubierta corresponde a la dirección del ángel, y el ángel puede inspeccionar entonces esos contenidos y seguir el procesamiento. En resumen, para ángeles ligados a procesos (es decir, ángeles que solo se aplican a los procesos autorizados en vez de itinerar de forma aleatoria por el estado de la PI), es deseable que el arcángel interprete las estructuras de datos de la PI.

Una función ejemplar adicional incluye proporcionar un canal asegurado de IPC que solo permitirá que vean los datos la aplicación de la PI y el ángel de la PD. El núcleo de la PI podría normalmente ver todas las páginas que atraviesa un canal de IPC entre la PI y la PD, pero si estos páginas solo pueden ser objeto de acceso bajo el ojo vigilante del arcángel, entonces se proporciona mucha garantía de que solo el proceso en cuestión (el proceso controlado por el ángel dado) pueda ver los datos del canal. Otra función ejemplar da al ángel la capacidad de controlar qué módulos (por ejemplo, DLL) y qué versiones de esos módulos pueden ser cargados en el espacio de proceso de un proceso dado.

Como entidad de confianza, el arcángel 380 tiene acceso a la memoria asociada con la PI y se le notifica en cualquier momento de algo que sucede en la PI. El arcángel 380 está programado de antemano con un cuerpo de conocimiento que usa para detectar incoherencias para determinar si debería tomarse alguna acción en interés de la seguridad o la protección. Por ejemplo, el arcángel 380 puede atrapar ciertas conjuntos de eventos de la PI. Estos pueden ser eventos que son permitidos por la PI y que no son excluidos por el nexa 351 y el entorno de confianza que gestiona. Por ejemplo, el arcángel 380 puede detectar correspondencias indebidas en la PI (que el nexa 351 permitiría en otro caso) que indiquen un posible ataque o un problema de seguridad. El arcángel 380 puede también llevar a cabo una comprobación de coherencia.

Para la realización mostrada en la Figura 3, cada ángel está limitado o supervisado de otra manera por el arcángel 380 y el nexa 351 (bloque 420). El arcángel 380 impone la unión entre un ángel y su código asociado de la PI, que limita la capacidad de los ángeles para afectar, por ejemplo, a la privacidad y la seguridad en la PI.

Es deseable que el comportamiento de los ángeles esté limitado a afectar únicamente a los procesos a los que se supone que están unidos, porque el nexo 351 y el arcángel 380 solo ejecutarán cualquier ángel que el usuario les indique que ejecuten según las directrices del usuario. El arcángel tiene poderes a la par que el nexo, y será objeto de escrutinio a aproximadamente el mismo nivel. Para los ángeles, como para cualquier otro agente, el nexo 5 ejecutará cualquier cosa que el usuario les diga. Por ello, aunque el nexo y los arcángeles están limitados, los ángeles ordinarios (como los agentes) no lo están (aunque el usuario puede establecer directrices que digan al nexos que ejecute o que no ejecute, por ejemplo, agentes o ángeles firmados por un evaluador particular).

Es deseable que los ángeles estén limitados. Por ejemplo, no debe permitirse que un ángel con un bloque de firma que diga "ángel para un primer programa" use la memoria del SO base de la PI ni que use la memoria de otros 10 programas. Permitir tal cosa violaría muchos derechos de usuario y haría que los ángeles fueran peligrosos en vez de útiles. Por ello, el arcángel se encarga de que solo los ángeles tengan acceso a los programas de la PI a los que se supone que pueden acceder.

Preferentemente, un agente de confianza no tiene más poder que cualquier programa de la PI. En particular, un agente de confianza no puede examinar el SO de la PI ni editar el estado de configuración del SO de la PI. En vez 15 de ello, a los ángeles, preferentemente, solo se les permite que inspeccionen o modifique la memoria de los mortales a los que se aplican. Además, en algunas realizaciones, el arcángel podría rechazar que un ángel cambie el código del mortal, restringiendo que el ángel lea nada en el espacio de direcciones del modo de usuario de su mortal y permitiéndole que escriba al espacio de memoria de lectura-escritura no compartido del mortal. Sin embargo, algunos mecanismos requieren que un mortal llame al ángel para que se le permita no volver al punto de 20 llamada, sino, más bien, a un punto de retorno calculado. Esto permite que el ángel obligue a algunos eventos a iniciarse en direcciones correctas conocidas en el mortal, una manera fuerte de combatir ataques de trampolín basados en pilas corrompidas que alteran direcciones de retorno.

Un ángel solo puede monitorizar su entidad o grupo de entidades asociados (bloque 430) y no cuenta con más confianza que cualquier otro agente. Un ángel no puede monitorizar ni mirar de otra forma entidades no asociadas. 25 En particular, un ángel tiene una o más de las siguientes propiedades:

- a. El ángel puede monitorizar la memoria del modo de usuario de solo el proceso o los procesos a los que está unido (es decir, el mortal) (un poder que normalmente no se otorga al código de la PD; véase más arriba).
- b. Solo el arcángel puede ver la memoria del modo de núcleo del SO de la PI al cual está unido.
- c. El ángel puede ser aplicado únicamente a aquellos procesos de la PI que puedan invocarlo o preguntar por él o se aplica únicamente a los procesos de la PI que él inicie.
- d. El ángel puede ser limitado por imposición declarativa. Por ejemplo, el nexos y/o el arcángel pueden obligar al ángel a que se proyecte solo a aquellos procesos que contengan ejecutables que coincidan con los ejecutables declarados en el manifiesto del ángel. Así, por ejemplo, un ángel para "herramienta intrusa" no 35 puede proyectarse a una aplicación de la PI por accidente ni por malicia sin que alguien cambie el manifiesto del ángel. Tal cambio del manifiesto sería obvio para una herramienta de directrices.

El arcángel 380 puede imponer las restricciones anteriores (bloques 440 y 450). Con este fin al arcángel puede dársele amplio acceso a la PI y, en ese caso es sometido a un nivel de escrutinio similar al del nexos (es decir, un escrutinio intenso). Por ejemplo, el arcángel tiene poder sobre el SO de la PI y, así, sobre cualquier cosa que se 40 ejecute en la PI. Dicho de otra forma, el arcángel puede leer cualquier memoria de la PI, pero no tiene poderes especiales de la PD, como acceso a la memoria del núcleo de la PD, ni capacidad de ver el interior de otros procesos de agente, ni restringir, aumentar, modificar, etc., el nexos u otros agentes de la PD. Un ángel solo puede leer el espacio de direcciones del programa al que se aplica (por ejemplo, los ángeles tienen poderes especiales que se aplican solo a los mortales a los que se aplican). El arcángel también puede leer toda la memoria de la PI (etapa 440) mientras ofrece servicios específicos de procesos para que los ángeles solo pueden ver el interior del espacio 45 de direcciones de los programas que monitorizan y protegen.

Un ángel puede "proyectarse" a su protegido en al menos una de las maneras siguientes (etapas 430 y 450):

- a. Puede bloquear o marcar como de solo lectura elementos de memoria, posiblemente en coordinación con comportamiento de protegido, para evitar ciertos cambios (por ejemplo, ataques de virus) al protegido.
- b. Puede llevar a cabo algunas operaciones clave para el protegido dentro de su espacio de confianza.
- c. Puede insistir en protecciones específicas del protegido, como limitar qué cambios de configuración pueden realizarse, o permitir que tales cambios se realicen si están autorizados por un ser humano autorizado que use un mecanismo seguro de entrada.

- d. Puede barrer la memoria y el estado del protegido a intervalos deseados buscando errores de coherencia, corrupciones, etcétera, y advertir al usuario o detener al protegido antes de que ocurran más daño o acción no deseada o no autorizada.
- 5 e. Puede entregar datos cifrados/codificados al protegido solo en la medida necesaria, para minimizar la cantidad de tales datos que pueda ser atacada en cualquier momento. 1. Puede usar un almacenamiento cifrado para mantener secretos cifrados para la PI (o una aplicación de la PI) y negarse a dar esos secretos a cualquier PI (o aplicación de la PI) que no tenga un compendio que o bien coincida con el propietario del secreto o esté enumerado como disponible por el propietario del secreto.
- 10 f. Dada una API apropiada, puede alterar el estado de ejecución del protegido; es decir, puede dirigir hilos a puntos de ejecución conocidos, redirigir el flujo de control en la aplicación objetivo, o llevar a cabo un cálculo y una ejecución de derivación para la aplicación objetivo. También puede editar el estado de configuración, el estado de arranque o similares, para forzar las cosas a modos aceptables para la operación segura/correcta del protegido.
- 15 g. Un ángel puede llamar al arcángel y pedir al arcángel para llevar a cabo la prevención, la protección, el descubrimiento o la reacción en nombre del protegido.
- h. Un ángel puede extraer (por ejemplo, por llamamiento o por inspección de la memoria) datos de salida de la aplicación, validar tales datos (por ejemplo, realizando una suma de control, etc.) y luego presentar estos datos usando un soporte físico seguro de salida.

20 Parte de la funcionalidad de la entidad o la aplicación puede ser movida al interior del ángel. De modo similar, parte de la funcionalidad del núcleo de la PI puede ser movida al arcángel. Un creador de aplicaciones puede implementar algunas de las funciones de aplicación del ángel. Aunque esto aumentaría la carga de la PD, permitiría que las funciones transferidas se efectuasen en el entorno de confianza. De forma similar, puede moverse en el arcángel 380 una porción del SO 301 de la PI.

25 Un ángel puede ser cargado o invocado de varias formas. Un programa de la PI, como la aplicación 305, puede invocar a su ángel 390. De esta manera, por ejemplo tras el arranque de una aplicación, se carga el correspondiente ángel. De forma alternativa, desde la PD puede invocarse un ángel, y el ángel invoca entonces al proceso o la aplicación correspondientes de la PI. El ángel usa al arcángel para pasar una llamada a la PI y solicitar que se arranque la aplicación. El arcángel une entonces el agente a la aplicación. Para una realización, las API que el nexa y el arcángel ofrecen al ángel de la aplicación le dejan ver únicamente el proceso que crea, y quizás los hijos del mismo.

30 Como alternativa adicional, el programa de la PI puede ser invocado por el manifiesto y luego desviado a la PD que arranca el ángel, que vuelve a llamar a la PI para que arranque el proceso o la aplicación correspondientes de la PI. Típicamente, un programa de la PI se arranca nombrando el fichero que lo contiene (una API que sea, por ejemplo, "run c:\algundi\algunotrodir\algunprograma.exe". Para un código de la PD (un agente o un ángel), se arranque nombrando un manifiesto, y el manifiesto nombra el binario. Esto es independiente de la ubicación. Además, los manifiestos típicamente están, por ejemplo, firmados y certificados, de modo que son mucho más difíciles de burlar. Así, un mecanismo ejemplar sería presentar un manifiesto combinado izquierdo/derecho a la PD (nexa) que arrancaría tanto la aplicación de la PI como el ángel relacionado y los uniría entre sí. Además, puede usarse el ángel para arrancar la aplicación ya sea desde la PI o la PD.

40 En una realización de la invención, el arcángel puede confirmar que la imagen del código cargado inicialmente del proceso de la PI coincide con una imagen del código objetivo declarado asociada con el ángel. La imagen del código objetivo declarado puede ser proporcionada por medio del manifiesto del ángel. Esto evita que el código que pretende ser un ángel para una aplicación particular arranque, en vez de ella, otra aplicación que proporcione una seguridad adicional contra un ataque.

45 Según algunas realizaciones de la invención, se evita que un ángel edite la imagen del código de la aplicación o el proceso de la PI con los que está asociado. El ángel puede leer/escribir datos, pero solo puede leer código.

Pueden emplearse estas y similares directrices para evitar que los ángeles se ejecuten sin supervisión o restricciones sobre la PI y se evita que ángeles maliciosos burlen el uso de programas y aplicaciones de la PI.

50 Además de los mecanismos de iniciación descritos en lo que antecede, hay otras maneras de garantizar que se una el ángel debido a la aplicación debida de la PI (o la PD) y que siga unido a la misma. Una aplicación que se esté ejecutando puede ser alterada por un atacante antes de efectuar una llamada a su agente o un virus de la PI puede interceptar y permutar su llamada para seleccionar algún otro ángel.

55 Las realizaciones de la presente invención pueden abordar esto procesando las llamadas desde una aplicación a su ángel a través de una autoridad de confianza, como el arcángel o el nexa. Por ejemplo, el arcángel puede compendiar la aplicación llamante de la PI y comparar el compendio con una lista de compendios "autorizados"

asociados con el ángel de la PD. Si no coinciden, ya sea porque la aplicación de la PI ha sido permutada, o porque la llamada ha sido modificada para que seleccione un ángel diferente, la llamada fracasa y el sistema puede advertir al usuario y/o tomar una cualquiera de varias acciones.

5 Puede usarse una directriz del sistema para especificar qué ángeles pueden unirse a qué aplicaciones de la PI. Usar un mecanismo estricto de directrices proporciona un mecanismo difícil de burlar y difícil de inicializar indebidamente para configurar tales dependencias.

En algunas realizaciones, un ángel tiene, preferentemente, niveles de inspección ajustables o programables diversos para enfrentarse a amenazas a la aplicación asociada. La sensibilidad del ángel a la amenaza o al ataque percibidos puede ser calibrada.

10 Además de proporcionar proyección (por ejemplo, defensa, custodia, consejo) al SO o las aplicaciones de la PI, podría aplicarse también un ángel a un agente que se ejecuta en el entorno informático de confianza. En tal caso, un agente objetivo (normalmente una entidad paranoide) confía en el ángel al que se une. Esto permite que un proceso externo de vigilancia evite diversas anomalías y vulnerabilidades en el agente objetivo. El ángel puede imponer invariantes de seguridad en vez de hacer un barrido en busca de errores de seguridad (por ejemplo, como en la tecnología antivirus convencional) y el uso de una separación rígida y una protección de procesos que proporciona un nexo.

20 Para una realización, el agente es una máquina virtual que presenta una "copia duplicada efectivamente idéntica" de alguna máquina real en la que se ha lanzado una imagen de un SO. Un entorno de confianza puede permitir que un agente acceda a la memoria de procesos de la máquina virtual. El agente que accede puede monitorizar la memoria del proceso para proteger a la máquina virtual contra ataques procedentes de la imagen que contiene. Un entorno de confianza puede permitir que un ángel proyecte la imagen del SO en la máquina virtual y permitir que los ángeles proyecten aplicaciones en la máquina virtual. Se contempla que los mismos mecanismos normalmente aplicados a las aplicaciones de la PI se apliquen en su lugar al entorno de la máquina virtual.

25 Para una realización de la invención, el nexo dota al arcángel de una API para la inspección y la alteración (al menos) de la memoria. El soporte de la API para atrapar los tentativas por cambiar las estructuras de control y reaccionar a las mismas facilita la proyección. Por ejemplo, en la arquitectura x86, puede proporcionarse la protección de estructuras de control, como GDT, LDT, IDT, registros de depuración, TR, etc., a través de una API. GDT se refiere a la tabla de descriptores globales, y LDT se refiere a la tabla de descriptores locales. Bloquear el GDTR (registro de la tabla de descriptores globales) detiene los ataques que dependen de la desviación del significado de direcciones virtuales para permitir saltos a lugares a los que el atacante normalmente no podría saltar. IDT es la tabla de despacho de interrupciones que controla el encaminamiento de interrupciones. La ubicación de la IDT es indicada por el IDTR (registro de la tabla de despacho de interrupciones). Bloquear el IDTR hace que la proyección sea más potente al detener ataques en los que el atacante usa la IDT y una interrupción publicada para forzar una derivación a código que no alcanzaría de otra manera.

35 Es deseable que el Entorno de Confianza (es decir, la PD) y el Entorno Abierto (es decir, la PI) estén conectados de alguna manera. La conexión permite que el Entorno de Confianza examine el estado y sea informado de eventos en el Entorno Abierto. Las enseñanzas aquí funcionan para las estructuras que incluyen, sin limitar de modo alguno, las estructuras siguientes:

- 40 1. La PD y la PI están en la misma máquina, y la PD puede examinar directamente la memoria de la PI (mientras que la PI no puede examinar la memoria de la PD sin permiso).
- 45 2. La PD y la PI están en procesadores diferentes, posiblemente con memorias diferentes, pero un bus, una red, un puerto u otra interconexión permiten que la PD vea el interior de la memoria de la PI. Por ejemplo, un procesador de servicio ARM podría ejecutar una pila de toda confianza, y la pila de confianza podría ser capaz de inspeccionar la memoria principal de un sistema MP x86. Por ejemplo, podrían tenerse una máquina con procesadores principales x86 y un ARM o un PowerPC como procesador de servicio, y usar los mecanismos de la presente invención para permitir que el procesador de servicio vigile el soporte lógico de los procesadores principales.
- 50 3. Si la PD puede recibir notificación de los eventos de la PI (por ejemplo, cambios de correspondencias), pero no alterarlos ni evitarlos, o no puede ver el interior de la memoria de la PI, sigue siendo posible alguna parte de proyección (por ejemplo, una parte débil).
- 55 4. La PD puede inspeccionar la memoria de la PI libremente, puede controlar (es decir, evitar o alterar) las modificaciones de la PI a la correspondencia de la memoria de la PI y direccionar estructuras de traducción, controlar el lugar al que apunta el vector de despacho de interrupciones (pero no precisa controlar el controlador de interrupciones, aunque, si se ofrece tal control, hay margen para eso). Se contempla que la determinación de una lista de estados/eventos que la PD pueda controlar plenamente de manera deseable para soportar una proyección fuerte es una tarea que debe hacerse para cada arquitectura de

procesadores, y un experto en la técnica comprenderá que la lista es diferente para arquitecturas diferentes.

5. En una realización, los cambios del registro TR x86 y la configuración de los registros de depuración del soporte físico también son controlables por la PD.

- 5 En el soporte físico de la técnica anterior, no se garantiza que se ejecute el entorno de confianza, porque puede depender de un soporte físico de interrupciones comunes, de la tabla de despacho de interrupciones de la PI, etcétera.

En el soporte físico enumerado más arriba, poder controlar la IDT (en un x86, o la equivalente en otros casos) permite que la PD garantice que alguna interrupción de su elección siempre ejecutará código que llame a la PD.

- 10 Sin embargo, un atacante o un error de la PI podrían corromper el controlador de interrupciones, desactivar las interrupciones, etcétera. Se contempla que el ATC (control de traducción de direcciones) se use para garantizar que la PD consiga ejecutarse de vez en cuando. Si la PD está usando el ATC, puede modificar el ATC para incrementar un contador. El contador se fija a algún valor siempre que la PD programa el arcángel. Si el contador llega a cero, el AT sabe entonces que el arcángel no se ha ejecutado durante "demasiado tiempo" y llama a un punto de entrada al
- 15 nexo que ejecuta el arcángel por la fuerza. Esta técnica no garantiza que el arcángel se ejecute en ningún momento particular, sino sí garantiza que se ejecutará después de un número de operaciones de edición de la memoria de la PI. Así, una PI que esté activa acabará teniendo que permitir que el arcángel se ejecute.

Si la PD puede bloquear la IDT y el sistema tiene una fuente fiable de NMI (interrupciones no enmascarables), entonces la PD puede obligar al gestor de NMI a llamar a la parte derecha.

- 20 En una realización ejemplar, el soporte físico tiene un temporizador que fuerza una interrupción a la PD después de cierto número de señales cronométricas.

La presente invención proporciona mecanismos que permite que la fiabilidad de un entorno informático sea proyecto a un segundo entorno informático. Dos o más sistemas operativos que se ejecutan en una sola máquina son ejemplos de "entornos" que pueden necesitar interactuar entre sí en una sola máquina, aunque se entenderá que la invención no está limitada a un sistema operativo tradicional. Además, al menos algunas de las técnicas descritas en el presente documento pueden ser usadas, en el caso general, para proyectar fiabilidad desde cualquier tipo de entidad ejecutable (por ejemplo, cualquier soporte físico) hasta cualquier otro tipo de entidad.

- 25 En el caso en el que dos entidades existen lado a lado en una sola máquina y necesitan interactuar entre sí, la interacción puede adoptar diversas formas. Por ejemplo, las dos entidades pueden precisar comunicarse datos entre sí en ambos sentidos. En el caso en el que las entidades son sistemas operativos (o ciertos otros tipos de entornos de ejecución, como motores de ejecución de secuencias de órdenes que ejecuten secuencias de órdenes en una máquina virtual), las entidades pueden precisar interactuar entre sí de ciertas otras maneras; por ejemplo, compartiendo memoria, compartiendo tiempo en un procesador, compartiendo recursos y gestionando interrupciones. La invención proporciona técnicas mediante las cuales dos entidades puede ocuparse de estos tipos
- 35 de interacciones mutuas mientras permiten que una entidad proyecte su fiabilidad a la otra entidad.

Las realizaciones descritas en lo que antecede se centran en la memoria como el recurso monitorizado, pero la invención no está limitada en ese sentido. Si hay disponibles monitores de seguridad para recursos distintos de la memoria, un agente base de monitorización (por ejemplo, un arcángel) puede emplear tales monitores como delegados de confianza para extender su esfera de confianza. Por ejemplo, si hay disponible una NIC segura, el agente base de monitorización puede usarla para excluir el envío de paquetes con ciertas cabeceras. En general, tal delegado de confianza solo precisa entender una invariante de medición, por ejemplo cabeceras que coincidan con <regexp>, y alertar de manera fiable al agente de monitorización de cambios de la invariante.

- 40 Se hace notar que los ejemplos precedentes se han proporcionado meramente con el fin de la explicación y que de ninguna manera debe interpretarse que limiten la presente invención. aunque la invención ha sido descrita con referencia a diversas realizaciones, se entiende que las palabras que se han usado en el presente documento son palabras de descripción e ilustración, no palabras de limitaciones. Además, aunque la invención ha sido descrita en el presente documento con referencia a medios, materiales y realizaciones particulares, no se pretende que la invención esté limitada a los particulares dados a conocer en el presente documento; más bien, la invención se extiende a todas las estructuras, los procedimientos y los usos funcionalmente equivalentes, como los que estén dentro del alcance de las reivindicaciones adjuntas. Los expertos en la técnica, teniendo el beneficio de las enseñanzas de esta memoria, pueden efectuar numerosas modificaciones a la misma, y pueden realizarse cambios sin apartarse del alcance y el espíritu de la invención en sus aspectos.
- 50

Lo que sigue es una lista de realizaciones preferentes adicionales de la invención:

Realización 1: Un sistema que comprende:

- 55 un entorno sin confianza;

un entorno de confianza; y

al menos un agente base de monitorización que se ejecuta en el entorno de confianza que monitoriza el entorno sin confianza.

5 Realización 2: El sistema de la realización 1 que, además, comprende una pluralidad de agentes de monitorización que se ejecutan en el entorno de confianza, monitorizando cada agente de monitorización al menos una aplicación, una extensión o un componente que se ejecutan en el entorno sin confianza.

10 Realización 3: El sistema de la realización 2 en el que cada agente de monitorización está asociado con una aplicación y en el que cada agente de monitorización monitoriza su aplicación asociada en busca de ataques e incoherencias, proyectando por ello la fiabilidad del entorno de confianza al entorno sin confianza.

Realización 4: El sistema de la realización 3 en el que cada agente de monitorización puede comprender una porción de la aplicación con la que está asociado.

Realización 5: El sistema de la realización 3 en el que cada agente de monitorización tiene un nivel ajustable de inspección para enfrentarse a amenazas a la aplicación asociada.

15 Realización 6: El sistema de la realización 3 en el que cada agente de monitorización puede recibir una entrada segura y transferir la entrada segura a la aplicación asociada.

Realización 7: El sistema de la realización 2 en el que al menos uno de los agentes de monitorización monitoriza agentes de aplicación que se ejecutan en el entorno de confianza.

20 Realización 8: El sistema de la realización 2 que, además, comprende otro agente de monitorización que se ejecuta en el entorno de confianza, en el que los agentes de monitorización están en comunicación mutua.

Realización 9: El sistema de la realización 1 en el que el agente base de monitorización detecta incoherencias en el entorno sin confianza.

Realización 10: El sistema de la realización 9 que, además, comprende una pluralidad de agentes de monitorización que detectan incoherencias en las aplicaciones que se ejecutan en el entorno sin confianza.

25 Realización 11: El sistema de la realización 10 que, además, comprende agentes de monitorización adicionales que detectan incoherencias en aplicaciones que se ejecutan en el entorno de confianza.

Realización 12: El sistema de la realización 1 en el que al menos uno de los agentes base de monitorización autoriza o desautoriza un evento del entorno sin confianza.

30 Realización 13: El sistema de la realización 12 en el que al menos uno de los agentes base de monitorización comprende una entrada segura para recibir una entrada, autorizando o desautorizando el agente base de monitorización en base a la entrada recibida.

Realización 14: El sistema de la realización 1 en el que al menos uno de los agentes base de monitorización se niega a permitir cambios al entorno de confianza sin recibir autorización por medio de una entrada segura.

35 Realización 15: El sistema de la realización 1 en el que al menos uno de los agentes base de monitorización se niega a permitir cambios al entorno sin confianza a no ser que los cambios sean descritos por un paquete que esté firmado por una entidad autorizada.

40 Realización 16: El sistema de la realización 1 que, además, comprende un agente de monitorización que se ejecuta en el entorno de confianza, monitorizando el agente de monitorización al menos una aplicación, una extensión o un componente que se ejecutan en el entorno sin confianza, en el que el agente de monitorización usa un almacenamiento cifrado para guardar un secreto para un sistema operativo o una aplicación que reside en el entorno sin confianza.

45 Realización 17: El sistema de la realización 16 en el que el agente de monitorización se niega a revelar el secreto al sistema operativo o a la aplicación, a no ser que el sistema operativo o la aplicación contengan un compendio que coincida con el propietario del secreto.

Realización 18: El sistema de la realización 16 en el que el agente de monitorización se niega a revelar el secreto al sistema operativo o a la aplicación, a no ser que el sistema operativo o la aplicación estén en una lista de compendios que pueden leer el secreto.

50 Realización 19: El sistema de la realización 16 en el que el agente de monitorización usa una prueba predeterminada para determinar si una entidad legítima está solicitando el secreto.

- Realización 20: El sistema de la realización 19 en el que la prueba predetermina incluye el examen de las pilas de la entidad y garantizar que las pilas tienen contenidos legales de pila.
- 5 Realización 21: El sistema de la realización 1 que, además, comprende un agente de monitorización que se ejecuta en el entorno de confianza, monitorizando el agente de monitorización al menos una aplicación, una extensión o un componente que se ejecutan en el entorno sin confianza, en el que el agente de monitorización edita un estado del entorno sin confianza para hacerlo seguro o aceptable de otra manera.
- Realización 22: El sistema de la realización 21 en el que el estado comprende una configuración inicial o una opción de informes de error.
- 10 Realización 23: El sistema de la realización 1 en el que el agente base de monitorización pone a cero la memoria física que no pertenece a la configuración que se sabe que es buena del entorno sin confianza o del entorno de confianza.
- Realización 24: El sistema de la realización 1 en el que el entorno sin confianza comprende un sistema básico de entrada/salida (BIOS), un soporte lógico inalterable o un cargador.
- 15 Realización 25: El sistema de la realización 1 que, además, comprende un nexo para ejecutar el agente base de monitorización en el arranque.
- Realización 26: El sistema de la realización 1 que, además, comprende un contador en el entorno de confianza, usándose el contador para determinar si debería ejecutarse el agente base de monitorización.
- Realización 27: El sistema de la realización 26 en el que el contador cuenta el número de operaciones de edición de la memoria sin confianza.
- 20 Realización 28: Un procedimiento de monitorización de un entorno sin confianza que comprende:
- proporcionar el entorno sin confianza;
 - proporcionar al menos un agente base de monitorización en el entorno de confianza; y
 - monitorizar el entorno sin confianza en busca de ataques e incoherencias para proyectar la fiabilidad del entorno de confianza al entorno sin confianza.
- 25 Realización 29: El procedimiento de la realización 28 que, además, comprende:
- proporcionar una pluralidad de agentes de monitorización que se ejecutan en el entorno de confianza; y
 - monitorizar al menos una aplicación, una extensión o un componente que se ejecutan en el entorno sin confianza.
- Realización 30: El procedimiento de la realización 29 que, además, comprende:
- 30 asociar cada agente de monitorización con una aplicación; y
- monitorizar cada aplicación asociada en busca de ataques e incoherencias.
- Realización 31: El procedimiento de la realización 29 que, además, comprende la asociación de una aplicación con uno de los agentes de monitorización y la transferencia de una porción de la aplicación al agente de monitorización para que la porción resida en el entorno de confianza.
- 35 Realización 32: El procedimiento de la realización 29 que, además, comprende la asociación de una aplicación con el agente de monitorización y la calibración de un nivel de inspección en el agente de monitorización para enfrentarse a amenazas en la aplicación asociada.
- Realización 33: El procedimiento de la realización 29 que, además, comprende la asociación de una aplicación con el agente de monitorización y la recepción de una entrada segura en el agente de monitorización y la transferencia de la entrada segura a la aplicación.
- 40 Realización 34: El procedimiento de la realización 29 en el que los agentes de monitorización están en comunicación mutua.
- Realización 35: El procedimiento de la realización 28 que, además, comprende que el agente base de monitorización autorice o desautorice un evento en el entorno sin confianza.
- 45 Realización 36: El procedimiento de la realización 35 que, además, comprende que el agente base de monitorización reciba una entrada de una entrada segura.

- Realización 37: El procedimiento de la realización 28 que, además, comprende que el agente base de monitorización se niegue a permitir cambios en el entorno sin confianza sin recibir autorización por medio de una entrada segura.
- 5 Realización 38: El procedimiento de la realización 28 que, además, comprende que el agente base de monitorización se niegue a permitir cambios en el entorno sin confianza a no ser que los cambios estén descritos por un paquete que esté firmado por una entidad autorizada.
- Realización 39: El procedimiento de la realización 28 que, además, comprende:
- proporcionar una pluralidad de agente de monitorización que se ejecutan en el entorno de confianza; y
- 10 que uno de los agentes de monitorización use un almacenamiento cifrado para guardar un sistema operativo para un sistema operativo o una aplicación que reside en el entorno sin confianza.
- Realización 40: El procedimiento de la realización 39 en el que el agente de monitorización se niega a revelar el secreto al sistema operativo o a la aplicación a no ser que el sistema operativo o la aplicación tengan un compendio que coincida con el propietario del secreto.
- 15 Realización 41: El procedimiento de la realización 39 en el que el agente de monitorización se niega a revelar el secreto al sistema operativo o a la aplicación a no ser que el sistema operativo o la aplicación estén en una lista de compendios que pueden leer el secreto.
- Realización 42: El procedimiento de la realización 39 que, además, comprende el uso de una prueba predeterminada para determinar si una entidad legítima está solicitando el secreto.
- 20 Realización 43: El procedimiento de la realización 42 en el que la prueba predeterminada incluye el examen de las pilas de la entidad y garantizar que las pilas tienen contenidos legales de pila.
- Realización 44: El procedimiento de la realización 28 que, además, comprende:
- proporcionar una pluralidad de agentes de monitorización que se ejecutan en el entorno de confianza; y
- que uno de los agentes de monitorización edite un estado del entorno sin confianza para hacerlo seguro o aceptable de otra manera.
- 25 Realización 45: El procedimiento de la realización 44 en el que el estado comprende una configuración inicial o una opción de informes de error.
- Realización 46: El procedimiento de la realización 28 que, además, comprende que el agente base de monitorización ponga a cero la memoria física que no pertenezca a la configuración buena conocida del entorno sin confianza o del entorno de confianza.
- 30 Realización 47: El procedimiento de la realización 28 en el que el entorno sin confianza comprende un sistema básico de entrada/salida (BIOS), un soporte lógico inalterable o un cargador.
- Realización 48: El procedimiento de la realización 28 que, además, comprende la ejecución del agente base de monitorización en el arranque por medio de un nexo.
- 35 Realización 49: El procedimiento de la realización 28 que, además, comprende la determinación de si el agente base de monitorización debiera ejecutarse en respuesta a un contador.
- Realización 50: El procedimiento de la realización 49 en el que el contador cuenta el número de operación de edición de la memoria sin confianza.
- Realización 51: El procedimiento de la realización 28 que, además, comprende proporcionar una pluralidad de agente de monitorización que detectan incoherencias en las aplicaciones que se ejecutan en el entorno sin confianza.
- 40 Realización 52: El procedimiento de la realización 51 que, además, comprende proporcionar agentes adicionales que detectan incoherencias en aplicaciones que se ejecutan en el entorno de confianza.
- Realización 53: Un sistema que comprende:
- un entorno de confianza que tiene al menos un sistema operativo, un soporte lógico inalterable y un sistema básico de entrada/salida (BIOS);
- 45 un entorno sin confianza; y

al menos un agente de monitorización que se ejecuta en el entorno de confianza y asociado con uno del sistema operativo, el soporte lógico inalterable, el BIOS y una aplicación que se ejecuta en el entorno sin confianza.

5 Realización 54: El sistema de la realización 53 en el que el al menos un agente de monitorización comprende una pluralidad de agentes de monitorización, teniendo cada agente de monitorización un poder asociado.

10 Realización 55: El sistema de la realización 53 en el que el entorno de confianza se ejecuta en una primera arquitectura de procesador y el entorno sin confianza se ejecuta en una segunda arquitectura de procesador que, además, comprende un agente base de monitorización que se ejecuta en el primer procesador.

Realización 56: El sistema de la realización 53 en el que el entorno de confianza y el entorno sin confianza se ejecutan en el mismo procesador que, además, comprende un agente base de monitorización que se ejecuta en el entorno de confianza.

15 Realización 57: El sistema de la realización 53 en el que el entorno de confianza se ejecuta en un primer procesador, y el entorno sin confianza se ejecuta en un segundo procesador, siendo capaces los procesadores primero y segundo de ejecutar ya sea en un modo de confianza o en un modo sin confianza.

Realización 58: El sistema de la realización 53 que, además, comprende un nexo y un agente base de monitorización que residen en el entorno de confianza, estando enlazado, ligado o compilado el agente base de monitorización en el nexo.

20 Realización 59: El sistema de la realización 53 que, además, comprende un nexo y un agente base de monitorización que residen en el entorno de confianza, en el que el agente base de monitorización es un procesador en el modo de usuario que se ejecuta en el nexo.

25 Realización 60: El sistema de la realización 53 que, además, comprende un agente base de monitorización que reside en el entorno de confianza y desarrollado por y con y en el mismo entorno de desarrollo o uno relacionado como un sistema operativo del entorno sin confianza.

Realización 61: El sistema de la realización 53 que, además, comprende un agente base de monitorización que reside en el entorno de confianza, siendo parte el agente base de monitorización de una base informática de confianza para la evaluación de la seguridad.

30 Realización 62: El sistema de la realización 53 que, además, comprende un agente base de monitorización, residiendo una primera porción del agente base de monitorización en el entorno de confianza, y residiendo una segunda porción del agente base de monitorización en una máquina físicamente remota, y estando conectadas las porciones primera y segunda por un enlace seguro.

REIVINDICACIONES

1. Un sistema que comprende:

un agente (355, 360) que se ejecuta en un entorno de ejecución de confianza, siendo el entorno de ejecución de confianza un entorno de ejecución que comprende un sistema operativo seguro (351) y que tiene acceso a una porción aislada de la memoria de un ordenador, en el que la porción aislada de la memoria del ordenador es accesible solo para el entorno de ejecución de confianza;

un agente (390, 395) de monitorización asociado al agente y que se ejecuta en el entorno de ejecución de confianza, estando adaptado el agente de monitorización para proteger al agente contra un soporte lógico malicioso; y

un agente base (380) de monitorización que se ejecuta en el entorno de ejecución de confianza, estando adaptado el agente base de monitorización para monitorizar el agente de monitorización.

2. El sistema de la reivindicación 1 en el que el agente es una máquina virtual que contiene una imagen de un sistema operativo.

3. El sistema de la reivindicación 2 en el que el agente de monitorización monitoriza la memoria de proceso para proteger a la máquina virtual contra ataques desde la imagen del sistema operativo contenido en la máquina virtual.

4. El sistema de la reivindicación 2 que, además, comprende una aplicación (305, 310) asociada con el agente de monitorización, en el que el agente base de monitorización protege a la aplicación contra un soporte lógico malicioso.

Entorno informático
100

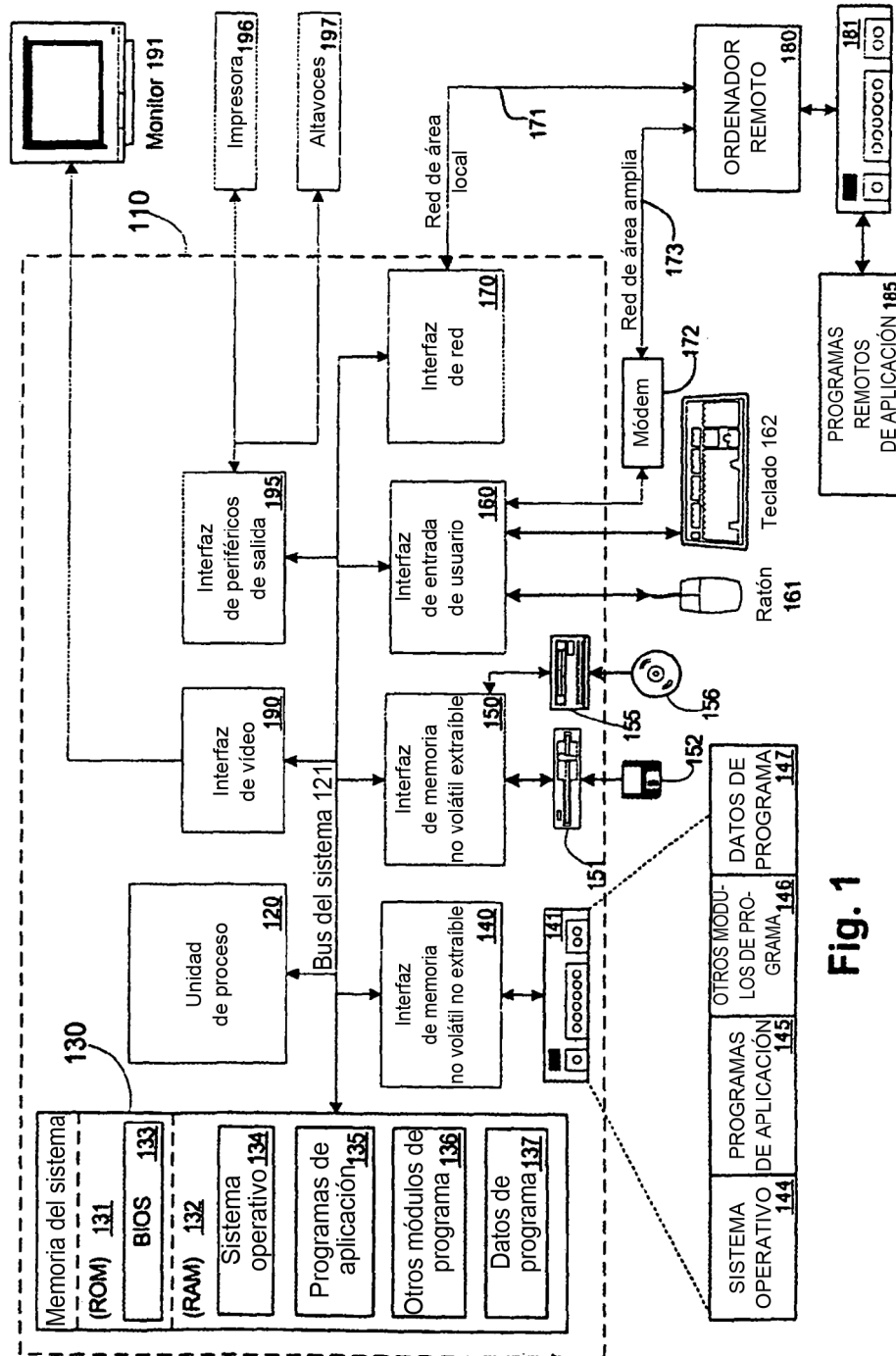


Fig. 1

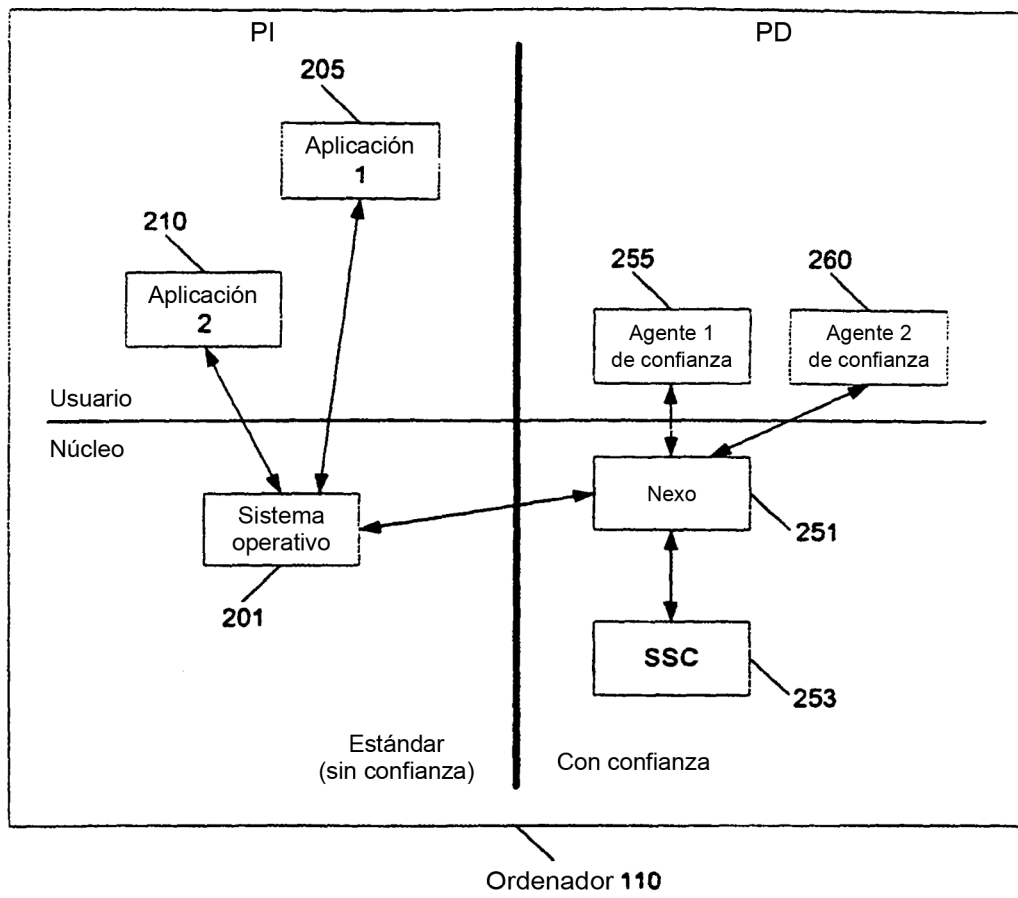


Fig. 2
(Técnica anterior)

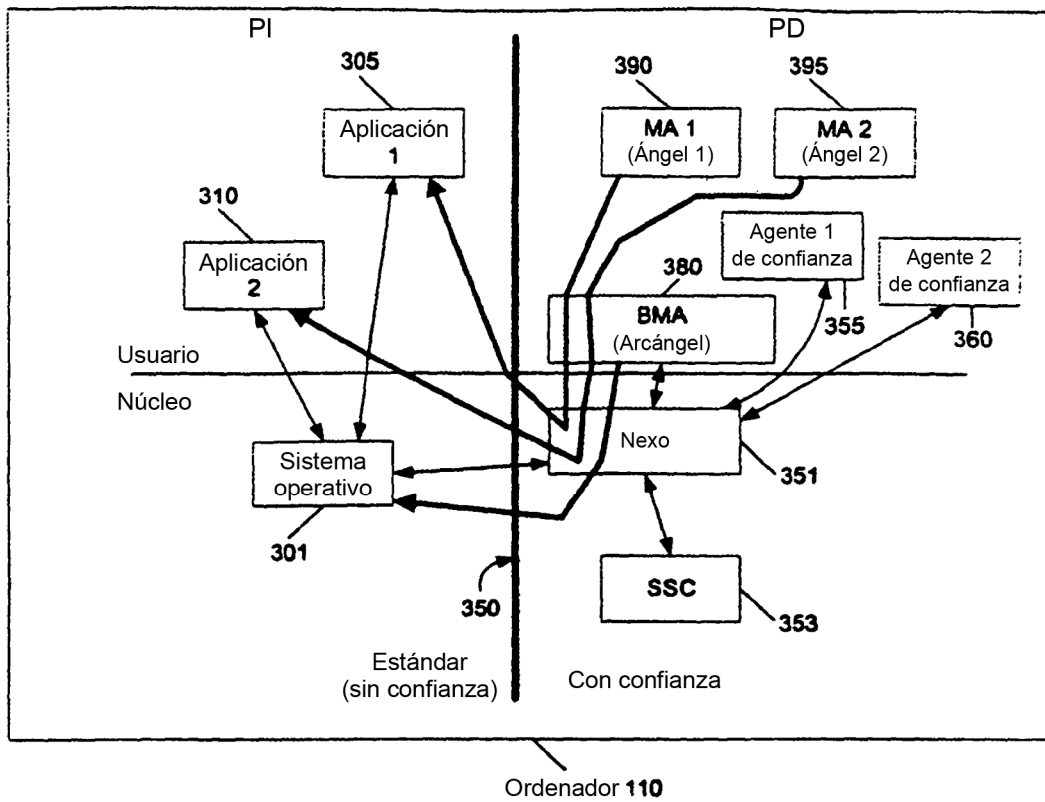


Fig. 3

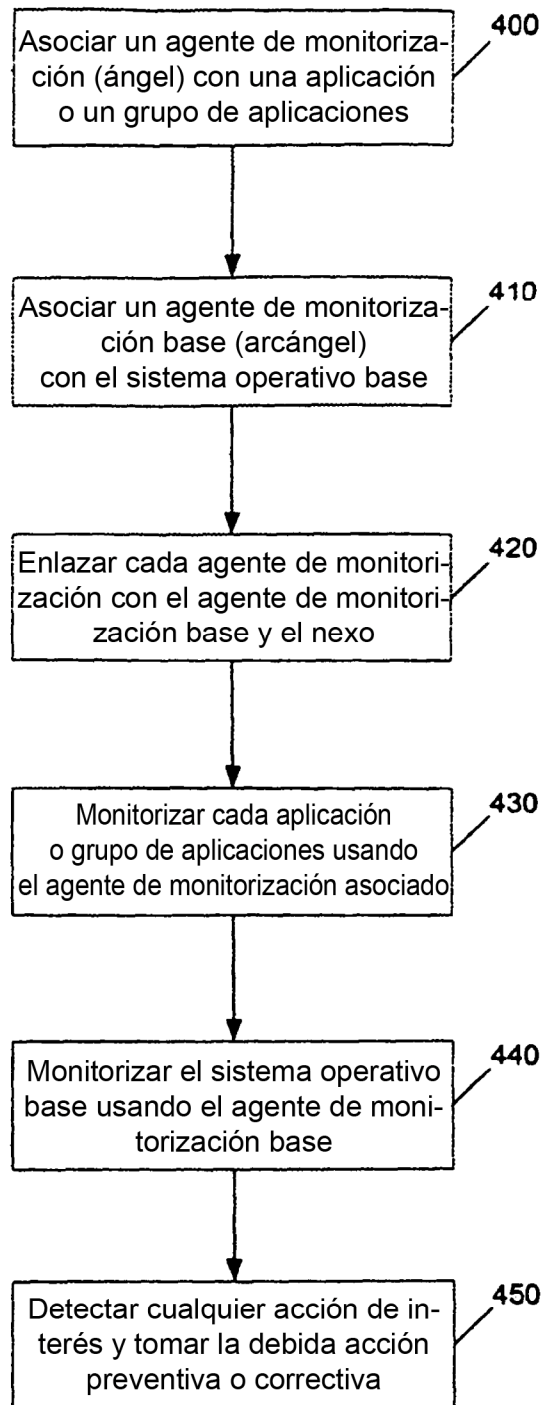


Fig. 4