(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 812 625**

(51) Int. CI.:

G06F 11/26 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **04.12.2015 PCT/EP2015/078671**

(87) Fecha y número de publicación internacional: **04.08.2016 WO16119953**

(96) Fecha de presentación y número de la solicitud europea: **04.12.2015 E 15805463 (5)**

(97) Fecha y número de publicación de la concesión europea: **15.07.2020 EP 3251012**

(54) Título: **Sistema de verificación para verificar un ordenador de un sistema informático en una red de verificación**

(30) Prioridad:

30.01.2015 DE 102015101388

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
17.03.2021

(73) Titular/es:

DEUTSCHE TELEKOM AG (100.0%)
Friedrich-Ebert-Allee 140
53113 Bonn, DE

(72) Inventor/es:

EGGERT, MARKUS y
HAUENSTEIN, DANIEL

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Sistema de verificación para verificar un ordenador de un sistema informático en una red de verificación

La presente invención se refiere a un sistema de verificación para verificar un ordenador de un sistema informático en una red de verificación y a un procedimiento para verificar un ordenador de un sistema informático en una red de verificación. La invención se refiere en particular a un procedimiento para automatizar pruebas o pruebas de seguridad en una red de comunicación y a un servidor de simulación (tarro de miel (*honeypot*)) para simular objetos de prueba.

Para vigilar el estado de un elemento de red, por ejemplo, de un servidor, de un enrutador, de un conmutador, etc. y, por ejemplo, verificar que esté configurado debidamente o que su versión de *software* esté actualizada, se han establecido en esencia dos métodos. Éstos son, por una parte, la utilización de una entidad central de verificación con acceso directo a todos los elementos de red y, por otra parte, la utilización de, así llamados, agentes, es decir, paquetes de *software* adicionales en todos los elementos de red que se comunican con una entidad central de verificación. Sin embargo, en los casos de un acceso directo de la entidad central de verificación a todos los elementos de red o de la instalación de agentes en los elementos de red, se presentan problemas en relación con la seguridad y la complejidad. Mediante el acceso directo a los elementos de red, la entidad de verificación o los agentes pueden ver y eventualmente manipular datos críticos en cuanto a la seguridad. La instalación de un agente en cada elemento de red plantea además grandes requisitos en cuanto a la complejidad, en particular en la verificación de grandes sistemas informáticos con un gran número de ordenadores o servidores.

El documento WO 2012/166591 divulga un sistema de verificación para verificar un ordenador, que constituye un objeto de prueba, de un sistema informático en una red de verificación.

El objetivo de la presente invención es crear un concepto para una verificación sencilla y segura de ordenadores de un sistema informático.

Este objetivo se logra mediante las características de las reivindicaciones independientes. Las reivindicaciones dependientes tienen por objeto formas de perfeccionamiento ventajosas.

Los dispositivos, procedimientos y sistemas presentados a continuación pueden ser de diferentes tipos. Los distintos elementos descritos pueden estar realizados mediante componentes de *hardware* o de *software*, por ejemplo, componentes electrónicos, que pueden producirse mediante diferentes tecnologías y que, por ejemplo, comprenden chips semiconductores, ASIC, microprocesadores, procesadores de señales digitales, circuitos eléctricos integrados, circuitos electroópticos y/o componentes pasivos.

Los dispositivos, procedimientos y sistemas presentados a continuación pueden utilizarse para detectar versiones de *software*, versiones de parches y configuraciones de ordenadores de un sistema informático. Un parche designa en este contexto una entrega de corrección para *software* o datos desde el punto de vista del usuario final, con el fin de cerrar brechas de seguridad, subsanar errores o reequipar funciones hasta el momento no existentes. En algunos productores, estas actualizaciones se denominan también *Service Pack*, cuando se componen de varios parches reunidos.

Según un primer aspecto, la invención se refiere a un sistema de verificación para verificar un determinado ordenador de un determinado sistema informático en una red de verificación, comprendiendo la red de verificación una entidad de verificación para verificar ordenadores del sistema informático, siendo el ordenador en cuestión un objeto de prueba y comprendiendo el sistema de verificación las características siguientes: un servidor de simulación, que está configurado para emular el objeto de prueba; y una entidad de control para controlar el servidor de simulación, estando la entidad de control configurada para ordenar al servidor de simulación que cree un objeto virtual de prueba para la emulación del objeto de prueba, estando la entidad de control diseñada además para ordenar a la entidad de verificación que, en lugar del ordenador en cuestión, verifique el objeto virtual de prueba creado por el servidor de simulación.

Tal sistema de verificación ofrece la ventaja de una verificación sencilla y segura de ordenadores de un sistema informático. Con el servidor de simulación es posible emular el objeto de prueba, de manera que, en lugar del objeto real de prueba, puede verificarse el objeto virtual de prueba. De este modo se eliminan problemas de seguridad existentes en el caso del acceso directo al ordenador, dado que no se verifica el ordenador real, sino el objeto virtual de prueba. Si la entidad de verificación parte de una amenaza de seguridad, ésta afecta al objeto virtual de prueba en el servidor de simulación, pero no al ordenador real mismo. Así pues, el servidor de simulación aísla el ordenador de la entidad de verificación y constituye un “tarro de miel”, es decir, el servidor de simulación sirve de medio de distracción para inducir a un *software* malicioso potencial o a virus presentes en la entidad de verificación a atacar o invadir el servidor de simulación en lugar del ordenador real. De este modo se aumenta la seguridad durante la verificación del sistema informático.

Además, tal servidor de simulación puede realizarse sin una gran complejidad. Gracias a que mediante el servidor de simulación puede verificarse cada ordenador de la misma manera, la entidad de verificación sólo necesita comunicarse con el servidor de simulación para implementar las tareas de verificación. Se suprime la instalación de paquetes de *software* propios en todos los ordenadores que se hayan de verificar, lo que, por ejemplo, sería necesario en caso de una realización de agentes para verificar los distintos ordenadores. Una vez terminada la verificación de un ordenador, puede borrarse de nuevo fácilmente el objeto virtual de prueba presente en el servidor de simulación y crearse el siguiente objeto virtual de prueba en el servidor de simulación.

Según el primer aspecto, el sistema de verificación está configurado para recibir datos del objeto de prueba y emular el objeto de prueba en el servidor de simulación basándose en los datos recibidos.

Esto ofrece la ventaja de que el objeto de prueba puede emularse basándose en datos del objeto de prueba, de manera que aquí puede hacerse una selección de los datos que sean relevantes para la emulación. De este modo, no es necesario reproducir la totalidad del objeto de prueba, es decir, el ordenador con todo el *software* instalado y todos los datos, sino que es suficiente con crear la emulación basándose en una selección de datos. Por ejemplo, el ordenador puede transmitir al servidor de simulación su *software*, parches y/o configuraciones instalados actualmente, de manera que el objeto virtual de prueba pueda reproducir el *software*, los parches, las configuraciones instalados en el ordenador.

En una forma de realización del sistema de verificación según el primer aspecto, los datos del objeto de prueba comprenden al menos una de las siguientes informaciones: versiones de *software* instalado en el objeto de prueba, versiones de parches instalados en el objeto de prueba, estados de configuraciones del objeto de prueba.

Esto ofrece la ventaja de que la emulación del ordenador puede crearse basándose en una selección de datos que sea justamente relevante para la entidad de verificación. El ordenador puede transmitir al servidor de simulación su *software*, parches y/o configuraciones instalados actualmente, de manera que el objeto virtual de prueba pueda reproducir el *software*, los parches, las configuraciones instalados en el ordenador. La selección puede ser realizada de manera controlada por la entidad de verificación.

Según el primer aspecto, el sistema de verificación está configurado para recibir los datos del objeto de prueba mediante una transferencia directa de archivos entre el objeto de prueba y el sistema de verificación o basándose en una interacción con un usuario del sistema de verificación.

Esto ofrece la ventaja de que los datos del objeto de prueba pueden llevarse de manera flexible al servidor de simulación. Por una parte es posible una transferencia automática de archivos, que puede ser iniciada tanto por el objeto de prueba como por el servidor de simulación. Por otra parte, es posible una interacción manual (o mediante voz o táctil) con un usuario, de manera que un explotador del sistema de verificación puede hacer que se verifiquen ordenadores especiales, por ejemplo, cuando para uno de éstos exista una sospecha relativa a la seguridad.

Según el primer aspecto, la entidad de control está configurada para verificar a intervalos regulares si existen datos de un nuevo objeto de prueba en el servidor de simulación y, en caso de existir los datos, ordenar al servidor de simulación que cree un objeto virtual de prueba para la emulación del nuevo objeto de prueba y ordenar a la entidad de verificación que verifique el objeto virtual de prueba.

Esto ofrece la ventaja de que los ordenadores del sistema informático pueden verificarse sucesivamente mediante una emulación de objetos virtuales de prueba correspondientes en el servidor de simulación. De este modo, es posible verificar un gran número de objetos de prueba mediante un solo servidor de simulación. Así pues, el sistema de verificación está realizado de manera que es eficaz y el gasto para verificar un gran número de objetos de prueba es pequeño. Además, para todos los objetos de prueba puede aplicarse el mismo procedimiento para crear el objeto virtual de prueba.

En una forma de realización del sistema de verificación según el primer aspecto, la entidad de control está configurada para, mediante una interfaz programable por aplicaciones, recibir datos y otros parámetros para el control y ordenar al servidor de simulación que cree un objeto virtual de prueba para la emulación del nuevo objeto de prueba y ordenar a la entidad de verificación que verifique el objeto virtual de prueba.

Esto ofrece la ventaja de que la entidad de control puede programarse o ajustarse de manera específica para el usuario y por lo tanto ofrece una gran flexibilidad.

En una forma de realización del sistema de verificación según el primer aspecto, la entidad de control está configurada para recibir de la entidad de verificación resultados de la verificación del objeto virtual de prueba.

Esto ofrece la ventaja de que todos los resultados de verificación están disponibles y pueden verse fácilmente en la entidad de control. Por ejemplo, la entidad de control puede entonces editar gráficamente los resultados de la verificación para representar los resultados de forma clara.

En una forma de realización del sistema de verificación según el primer aspecto, el sistema de verificación comprende una primera interfaz de verificación entre la entidad de control y la entidad de verificación, comprendiendo la primera interfaz de verificación una interfaz programable por aplicaciones.

Esto ofrece la ventaja de que mediante la primera interfaz de verificación es posible controlar o programar la entidad de verificación. De este modo, el sistema de verificación puede emplearse con diferentes entidades de verificación, que, por ejemplo, pueden proceder de diferentes productores.

En una forma de realización del sistema de verificación según el primer aspecto, el sistema de verificación comprende una segunda interfaz de verificación entre el servidor de simulación y la entidad de verificación, estando la segunda interfaz de verificación configurada para cargar *plug-ins* puestos a disposición por la entidad de verificación con el fin de verificar el objeto virtual de prueba en el servidor de simulación.

Esto ofrece la ventaja de que, mediante la segunda interfaz de verificación, la entidad de verificación puede controlar la verificación del objeto virtual de prueba en el servidor de simulación. Mediante los *plug-ins* puestos a disposición por la entidad de verificación es posible definir las tareas de verificación que tienen lugar en el servidor de simulación.

En una forma de realización del sistema de verificación según el primer aspecto, el servidor de simulación está configurado para, mediante los *plug-ins* puestos a disposición por la entidad de verificación, escanear el objeto virtual de prueba en busca de una versión de *software*, una versión de parche o un estado de configuración predeterminados.

Esto ofrece la ventaja de que una consulta de versiones de *software*, versiones de parche y/o configuraciones de los ordenadores del sistema informático puede realizarse rápida y eficazmente. De este modo, es posible formarse rápida y fácilmente una idea de las versiones de *software*/versiones de parches/el estado de configuración de grandes sistemas informáticos, como por ejemplo centros de conmutación con un gran número de servidores para conmutar enlaces de comunicación o enrutadores con un gran número de módulos de procesador para enrutar enlaces, etc.

En una forma de realización del sistema de verificación según el primer aspecto, el servidor de simulación dispone de datos de acceso al objeto de prueba, que son válidos sólo en el objeto virtual de prueba y sólo para el intervalo de tiempo de la verificación del objeto virtual de prueba.

Esto ofrece la ventaja de una seguridad elevada, dado que el sistema de verificación, que, por ejemplo, puede proceder de un proveedor externo y por lo tanto puede clasificarse como inseguro, no obtiene acceso directo al ordenador o al objeto real de prueba. Cuando los datos de acceso son válidos sólo en el objeto virtual de prueba, no puede vulnerarse la seguridad del objeto real de prueba. Cuando los datos de acceso son válidos sólo durante un intervalo de tiempo de verificación, puede excluirse un uso indebido en otros momentos.

Según un segundo aspecto, la invención se refiere a un procedimiento para verificar un determinado ordenador de un determinado sistema informático en una red de verificación, mediante un sistema de verificación, comprendiendo el sistema de verificación un servidor de simulación y una entidad de control, comprendiendo la red de verificación una entidad de verificación para verificar ordenadores del sistema informático, siendo el ordenador en cuestión un objeto de prueba, comprendiendo el procedimiento las etapas siguientes: crear un objeto virtual de prueba para la emulación del objeto de prueba en un servidor de simulación; ordenar a la entidad de verificación que, en lugar del ordenador en cuestión, verifique el objeto virtual de prueba creado en el servidor de simulación.

Tal procedimiento ofrece la ventaja de una verificación sencilla y segura de ordenadores de un sistema informático. Con la creación del objeto virtual de prueba en el servidor de simulación es posible emular el objeto de prueba, de manera que, en lugar del objeto real de prueba, puede verificarse el objeto virtual de prueba. De este modo se eliminan problemas de seguridad existentes en el caso del acceso directo al ordenador, dado que no se verifica el ordenador real, sino el objeto virtual de prueba. De este modo, el procedimiento aumenta la seguridad durante la verificación del sistema informático.

Además, con el procedimiento puede verificarse cada ordenador de la misma manera mediante el servidor de simulación. Una vez terminada la verificación de un ordenador, puede borrarse de nuevo fácilmente el objeto virtual de prueba presente en el servidor de simulación y crearse y verificarse el siguiente objeto virtual de prueba en el servidor de simulación. De este modo, el procedimiento puede realizarse fácilmente y sin un gran gasto.

En una forma de realización del procedimiento según el segundo aspecto, el procedimiento comprende además: escanear el objeto virtual de prueba en busca de al menos uno de los datos siguientes: versiones del *software* instalado en el objeto de prueba, de los parches instalados en el objeto de prueba, estados de configuraciones del objeto de prueba.

Esto ofrece la ventaja de que una consulta de versiones de *software*, versiones de parches y/o configuraciones de los ordenadores del sistema informático puede realizarse con el procedimiento rápida y eficazmente. De este modo, es

posible formarse rápida y fácilmente una idea de las versiones de *software*/versiones de parches/el estado de configuración de grandes sistemas informáticos.

Según el segundo aspecto, el procedimiento comprende además: verificar si existen datos del objeto de prueba en el servidor de simulación; en caso de existir datos del objeto de prueba: ordenar al servidor de simulación que cree un objeto virtual de prueba basándose en los datos del objeto de prueba; ordenar a la entidad de verificación que verifique el objeto virtual de prueba; recibir resultados de la verificación del objeto virtual de prueba; y borrar el objeto virtual de prueba en el servidor de simulación.

Esto ofrece la ventaja de que, por medio de estas operaciones, el procedimiento puede realizarse de una manera rápida y poco complicada. Estas cinco etapas consistentes en verificar, ordenar al servidor de simulación, ordenar a la entidad de verificación, recibir y borrar pueden realizarse para un número cualquiera de objetos de prueba sucesivos o eventualmente también paralelamente, para obtener rápidamente resultados de verificación.

Según un tercer aspecto, la invención se refiere a un programa informático para una interfaz programable por aplicaciones entre la entidad de control de un sistema de verificación según el primer aspecto y una entidad de verificación en una red de verificación, que comprende: un primer módulo de programa, que está configurado para transmitir a la entidad de verificación una identificación de un objeto virtual de prueba creado por el servidor de simulación; un segundo módulo de programa, que está diseñado para ordenar a la entidad de verificación que, en lugar del ordenador en cuestión, verifique el objeto virtual de prueba; y un tercer módulo de programa, que está diseñado para ordenar a la entidad de verificación que transmita a la entidad de control resultados de la verificación del objeto virtual de prueba.

Tal programa informático ofrece la ventaja de que crea una interfaz programable por aplicaciones para una entidad de verificación. De este modo puede utilizarse en el sistema de verificación cualquier entidad de verificación, realizándose la adaptación de la entidad de verificación al servidor de simulación mediante la interfaz programable por aplicaciones.

En una forma de realización del programa informático según el tercer aspecto, el programa informático comprende un cuarto módulo de programa, que está diseñado para ordenar a la entidad de verificación que escanee el objeto virtual de prueba en busca de al menos uno de los datos siguientes: versiones de *software* instalado en el objeto de prueba, versiones de parches instalados en el objeto de prueba, estados de configuraciones del objeto de prueba.

Esto ofrece la ventaja de que mediante la interfaz programable por aplicaciones es posible realizar rápida y eficazmente una consulta de versiones de *software*, versiones de parches y/o configuraciones de los ordenadores del sistema informático. Esto permite ordenar que se realicen y realizar rápida y eficazmente las tareas de verificación esenciales de la entidad de verificación.

Haciendo referencia a los dibujos adjuntos, se explican otros ejemplos de realización. Se muestran en:

La figura 1, una representación esquemática de un sistema 100 de verificación según una primera forma de realización;
la figura 2, una representación esquemática de un sistema 200 de verificación según una segunda forma de realización;
la figura 3, una representación esquemática de un procedimiento 300 para verificar un ordenador de un sistema informático según una forma de realización; y
la figura 4, una representación esquemática de un programa 400 de aplicación de una interfaz programable por aplicaciones entre una entidad de control y una entidad de verificación según una forma de realización.

En la descripción detallada siguiente se hace referencia a los dibujos adjuntos, que forman parte de la misma y en los que, a modo de ilustración, se muestran formas de realización específicas en las que la invención puede realizarse. Se entiende que también pueden utilizarse otras formas de realización y llevarse a cabo modificaciones estructurales o lógicas sin apartarse del concepto de la presente invención. Por lo tanto, la descripción detallada siguiente no debe entenderse en un sentido restrictivo. Además, se entiende que las características de los distintos ejemplos de realización descritos en la presente memoria pueden combinarse entre sí, siempre que no se indique específicamente otra cosa.

Los aspectos y las formas de realización se describen haciendo referencia a los dibujos, en donde los símbolos de referencia iguales se refieren en general a elementos iguales. En la descripción siguiente se exponen numerosos detalles con fines de explicación, para facilitar una comprensión a fondo de uno o varios aspectos de la invención. Sin embargo, para un experto en la técnica puede ser evidente que uno o varios aspectos o formas de realización pueden realizarse con un menor grado de los detalles específicos. En otros casos, se representan en forma esquemática estructuras y elementos conocidos para facilitar la descripción de uno o varios aspectos o formas de realización. Se entiende que pueden utilizarse otras formas de realización y llevarse a cabo modificaciones estructurales o lógicas sin apartarse del concepto de la presente invención.

Aunque una determinada característica o un determinado aspecto de una forma de realización puedan haberse divulgado en relación con sólo una de varias implementaciones, tal característica o tal aspecto pueden además combinarse con otra u otras características o aspectos de otras implementaciones, como se desee y pueda ser ventajoso para una aplicación dada o determinada. Además, en la medida en que se utilicen las expresiones "contener", "tener", "con" u otras variantes de las mismas en la descripción detallada o en las reivindicaciones, tales expresiones han de ser inclusivas de una manera similar a la expresión "comprender". Las expresiones "acoplado" y "conectado" pueden haberse utilizado junto con derivaciones de las mismas. Se entiende que tales expresiones se utilizan para indicar que dos elementos cooperan o interactúan entre sí independientemente de que estén en contacto físico o eléctrico directo o que no estén en contacto directo entre sí. Además, la expresión "a modo de ejemplo" debe interpretarse solamente como un ejemplo, en lugar de la denominación para lo mejor u óptimo. Por lo tanto, la descripción siguiente no debe entenderse en un sentido restrictivo.

La figura 1 muestra una representación esquemática de un sistema 100 de verificación según una primera forma de realización. El sistema 100 de verificación sirve para verificar un determinado ordenador 101 de un determinado sistema informático en una red de verificación. Es decir, puede seleccionarse un determinado ordenador de un sistema informático y someterse el mismo a una verificación. Además, puede seleccionarse un determinado sistema informático de una pluralidad de distintos sistemas informáticos y someterse el mismo a la verificación. La red de verificación comprende una entidad 103 de verificación para verificar ordenadores del sistema informático. El ordenador 101 en cuestión es en este contexto el objeto 101 de prueba.

El sistema 100 de verificación comprende un servidor de simulación 105 y una entidad 107 de control, que están acoplados mediante una interfaz 108 de control. El servidor 105 de simulación sirve para emular el objeto 101 de prueba. La entidad 107 de control sirve para controlar el servidor 105 de simulación. La entidad de control ordena al servidor 105 de simulación que cree un objeto virtual de prueba para la emulación del objeto 101 de prueba y ordena a la entidad 103 de verificación que, en lugar del ordenador 101 en cuestión, verifique el objeto virtual de prueba creado por el servidor 105 de simulación.

El sistema 100 de verificación recibe datos 111 del objeto 101 de prueba mediante una interfaz 102 de prueba entre el objeto de prueba y el sistema 100 de verificación y emula el objeto 101 de prueba basándose en los datos 111 recibidos. Los datos 111 del objeto 101 de prueba pueden ser versiones de *software* instalado en el objeto 101 de prueba, versiones de parches instalados en el objeto 101 de prueba, estados de configuraciones del objeto 101 de prueba u otros datos. La interfaz 102 de prueba puede ser tanto una simple interfaz de transferencia de archivos (basada en protocolos como FTP, SCP, SMTP, …) como una interfaz programable por aplicaciones, mediante la cual puedan entregarse no sólo datos de configuración, sino también otros parámetros para el control del sistema de verificación.

El sistema 100 de verificación puede recibir los datos 111 del objeto 101 de prueba mediante una transferencia directa de archivos entre el objeto 101 de prueba y el sistema 100 de verificación o sobre la base de una interacción con un usuario del sistema 100 de verificación (no representada en la figura 1). Si el sistema 100 de verificación recibe datos 111 de un objeto 101 de prueba, la entidad 107 de control ordena al servidor 105 de simulación que cree un objeto virtual de prueba para la emulación del nuevo objeto 101 de prueba y ordena a la entidad 103 de verificación que verifique el objeto virtual 109 de prueba. La entidad 107 de control puede recibir de la entidad de verificación resultados de la verificación del objeto virtual de prueba.

El sistema 100 de verificación comprende una primera interfaz 106 de verificación entre la entidad 107 de control y la entidad 103 de verificación. La primera interfaz 106 de verificación puede ser una interfaz programable por aplicaciones. El sistema 100 de verificación comprende una segunda interfaz 104 de verificación entre el servidor 105 de simulación y la entidad 103 de verificación. Mediante la segunda interfaz 104 de verificación pueden cargarse en el servidor 105 de simulación *plug-ins* puestos a disposición por la entidad 103 de verificación para verificar el objeto virtual 109 de prueba. El servidor 105 de simulación puede, mediante los *plug-ins* puestos a disposición por la entidad 103 de verificación, escanear el objeto virtual de prueba en busca de una versión de *software*, una versión de parche o un estado de configuración predeterminados.

El servidor 105 de simulación dispone de datos de acceso al objeto 101 de prueba. Estos datos de acceso pueden, por ejemplo, ser válidos sólo en el objeto virtual de prueba. Además, estos datos de acceso pueden ser válidos sólo para el intervalo de tiempo de la verificación del objeto virtual de prueba.

La figura 2 muestra una representación esquemática de un sistema 200 de verificación según una segunda forma de realización. El sistema 200 de verificación puede corresponder al sistema 100 de verificación descrito anteriormente en relación con la figura 1. La interfaz 102 de prueba puede estar implementada directamente entre el objeto 101 de prueba y el sistema 200 de verificación, de manera que un archivo 111a generado en el objeto de prueba pueda transmitirse, por ejemplo, mediante una transferencia T1 de archivos directamente al sistema 200 de verificación. Como alternativa, la interfaz 102 de prueba puede extenderse a través de un terminal de trabajo conectado entre el objeto 101 de prueba y el sistema 200 de verificación, de manera que un archivo 111a generado en el objeto de prueba pueda transmitirse, por ejemplo, mediante una primera transferencia T2a de archivos en primer lugar al terminal 117

de trabajo y después, por ejemplo, mediante una segunda transferencia T2b de archivos del terminal 117 de trabajo al sistema 200 de verificación. En el terminal 117 de trabajo, un usuario tiene la posibilidad de modificar el archivo 111a transmitido, de manera que el archivo 111 presente en el sistema 200 de verificación pueda eventualmente diferenciarse del archivo 111a generado en el objeto 101 de prueba.

La entidad 107 de control puede estar implementada en un ordenador, por ejemplo, en un servidor. También puede estar implementada en el servidor 105 de simulación mismo, de manera que la interfaz 108 de control constituya una interfaz interna. La entidad 107 de control puede además estar implementada como un *software* de aplicación (App) en un ordenador o procesador separado, por ejemplo un ordenador de control para controlar el servidor 105 de simulación.

En la figura 2 puede verse que el objeto virtual 109 de prueba se genera en el servidor 105 de simulación.

La figura 2 muestra el sistema 200 de verificación, en el que se utiliza una entidad activa central 103 de verificación, que no obstante accede sólo al servidor 105 de simulación que, utilizando datos 111 obtenidos previamente de los objetos 101 de prueba, simula los objetos 101 que se han de probar. Con este fin, se generan en el objeto 101 de prueba los datos 111a, que se transfieren al servidor 105 de simulación. Esto puede llevarse a la práctica mediante una transferencia directa T1 de archivos entre los objetos 101 de prueba y el servidor 105 de simulación o ser iniciado (T2a y T2b) manualmente por un usuario utilizando un terminal 117 de trabajo opcional. La entidad 107 de control verifica a intervalos regulares si hay nuevos archivos 111 en el servidor 105 de simulación (etapa P1 de proceso). Si es éste el caso, se crea en el servidor 105 de simulación un objeto virtual 109 de prueba que, utilizando los datos 111 del objeto 101 de prueba, simula el objeto 101 de prueba frente a la entidad de prueba o entidad 103 de verificación (etapa P2 de proceso). La entidad 107 de control activa entonces la entidad 103 de verificación y le ordena verificar el objeto virtual 109 de prueba (etapa P3 de proceso). Si en este contexto es necesaria una verificación interna, pueden utilizarse datos de acceso que sean válidos sólo en el objeto virtual 109 de prueba y sólo para el intervalo de tiempo de la prueba. Una vez terminadas las pruebas, la entidad 103 de verificación pone a disposición de la entidad 107 de control los resultados de la verificación (etapa 4 de proceso). Después, la entidad 107 de control termina la ejecución del objeto virtual 109 de prueba en el servidor 105 de simulación (etapa P5 de proceso).

Tal sistema 200 de verificación ofrece diversas ventajas en relación con el acceso directo de una entidad 103 de verificación a un objeto 101 de prueba, en particular cuando han de verificarse una pluralidad de objetos 101 de prueba. Asimismo, una entidad central, por ejemplo, en caso de utilizarse la entidad 103 de verificación como entidad central, que necesite acceder a muchos elementos de red, está siempre asociada con riesgos de seguridad, dado que, por una parte, necesita acceder por técnica de red a todos los elementos de red y, por otra parte, también ha de disponer frecuentemente de datos de acceso (por ejemplo, nombre de usuario y contraseña) del objeto 101 de prueba. Si la entidad 103 de verificación se viese comprometida, un atacante tendría de este modo acceso a un gran número de elementos de red. Este riesgo no existe en el sistema de verificación aquí presentado, dado que la entidad 103 de verificación no tiene acceso directo a los elementos de red, sino que el acceso se realiza mediante el sistema 200 de verificación con servidor 105 de simulación y entidad 107 de control.

En una forma de realización del sistema 200 de verificación, el servidor 105 de simulación constituye un tarro de miel (*honeypot*), es decir, un servidor que simula los servicios de red de un ordenador, de toda una red de ordenadores o el comportamiento de un usuario. El tarro de miel puede utilizarse, además de para aislar el objeto 101 de prueba en relación con un atacante, para obtener información sobre patrones de ataque y sobre el comportamiento de atacantes. Si se realiza un acceso a tal servicio o usuario virtual, es posible protocolizar todas las acciones asociadas con el mismo y en caso dado disparar una alarma. La valiosa red real o el valioso sistema informático real quedan libres dentro de lo posible de intentos de ataque, dado que están mejor protegidos que el tarro de miel o el servidor 105 de simulación. La idea detrás de los servicios tipo tarro de miel del servidor 105 de simulación es instalar en una red uno o varios tarros de miel que no ofrezcan servicios necesarios para el usuario mismo o sus compañeros de comunicación y por lo tanto nunca sean abordados durante el funcionamiento normal. Un atacante que no pueda diferenciar entre servidores o programas reales y tarros de miel y examine de forma rutinaria todos los componentes de red en cuanto a puntos débiles, recurrirá antes o después a los servicios ofrecidos por un tarro de miel y en este contexto será protocolizado por el tarro de miel. Dado que es un sistema no utilizado, todo acceso al mismo debe clasificarse como un posible intento de ataque. Así pues, el servidor 105 de simulación sirve al mismo tiempo tanto de objeto virtual de prueba, para posibilitar la verificación de los ordenadores del sistema informático, como de tarro de miel, para prevenir y protocolizar ataques a la red de ordenadores.

La entidad 103 de verificación puede ser un servidor específico del productor, mediante el cual se realice una verificación del sistema informático. Por ejemplo, la entidad 103 de verificación puede comprender un servidor Nessus. Nessus es un escáner de red y seguridad. Se basa en el principio cliente-servidor, es decir, se inicia en un ordenador del servidor Nessus y a continuación es posible conectarse a uno o varios clientes desde un ordenador o local o remoto. Esta operación se protege mediante certificados SSL y contraseñas. Con el arranque del servidor se cargan automáticamente los *plug-ins*. Con estos *plug-ins* pueden hallarse diversas brechas de seguridad del sistema operativo o de los servicios que se ejecutan en el anfitrión que se ha de escanear. Los *plug-ins* se crean en el lenguaje de secuencia de comandos (*script*) propio de Nessus “Nessus Attack Scripting Language” (NASL). Por medio del

programa de cliente se conecta uno a continuación al servidor y se establece una sesión (*session*), en la que es posible introducir o modificar los *plug-ins*, el anfitrión de destino y otros ajustes. Una vez ejecutado el escaneo en un anfitrión, el cliente Nessus emite una sinopsis de los puertos abiertos y eventuales brechas de seguridad halladas.

Por ejemplo, la entidad 103 de verificación puede comprender un servidor Qualys. Con tal servidor puede escanearse el sistema informático. Pueden descubrirse brechas de seguridad o potenciales de peligro y pueden ponerse a disposición los parches adecuados para subsanar las brechas de seguridad. Con el servidor Qualys pueden registrarse informes de escaneo interactivos, por ejemplo, por amenaza o por parche. Pueden probarse páginas web y aplicaciones (App) en cuanto a los riesgos principales y *software* malicioso. Además, pueden evaluarse ordenadores en relación con la seguridad, por ejemplo, en relación con SCAP (*Security Content Automation Protocol* (protocolo de automatización de contenido de seguridad)) u OWASP (*Open Web Application Security Project* (proyecto abierto de seguridad de aplicaciones web)). SCAP es un método para la utilización de determinados estándares para la evaluación automatizada de gestión de vulnerabilidad, de medición y de cumplimiento de políticas. OWASP es una organización de estandarización con el objetivo de mejorar la seguridad de aplicaciones y servicios en la *World Wide Web*.

La figura 3 muestra una representación esquemática de un procedimiento 300 para verificar un ordenador de un sistema informático según una forma de realización.

Con el procedimiento 300 puede verificarse un determinado ordenador 101 de un determinado sistema informático en una red de verificación, comprendiendo la red de verificación una entidad 103 de verificación para verificar ordenadores del sistema informático y siendo el ordenador 101 en cuestión un objeto 101 de prueba, como se ha descrito anteriormente con mayor detalle en relación con las figuras 1 y 2. El procedimiento 300 comprende las etapas siguientes: crear 301 un objeto virtual 109 de prueba para la emulación del objeto 101 de prueba en un servidor 105 de simulación, por ejemplo, de acuerdo con la descripción relativa a las figuras 1 y 2; y ordenar a la entidad 103 de verificación que, en lugar del ordenador 101 en cuestión, verifique el objeto virtual 109 de prueba creado en el servidor 105 de simulación, por ejemplo, de acuerdo con la descripción relativa a las figuras 1 y 2.

El procedimiento 300 puede comprender además la etapa siguiente: escanear el objeto virtual 109 de prueba en busca de al menos uno de los datos siguientes: versiones de *software* instalado en el objeto 101 de prueba, versiones de parches instalados en el objeto 101 de prueba, estados de configuraciones del objeto 101 de prueba, por ejemplo, de acuerdo con la descripción relativa a las figuras 1 y 2.

El procedimiento 300 puede comprender además las cinco etapas P1, P2, P3, P4, P5 siguientes, que, por ejemplo, pueden implementarse en la entidad 107 de control como se ha descrito anteriormente en relación con las figuras 1 y 2: verificar (P1) si existen datos 111 del objeto 101 de prueba en el servidor 105 de simulación; en caso de existir datos 111 del objeto 101 de prueba: ordenar al servidor 105 de simulación que cree (P2) un objeto virtual 109 de prueba basándose en los datos 111 del objeto 101 de prueba; ordenar a la entidad 103 de verificación que verifique (P3) el objeto virtual 109 de prueba; recibir (P4) resultados de la verificación del objeto virtual 109 de prueba; y borrar (P5) el objeto virtual 109 de prueba en el servidor 105 de simulación.

La figura 4 muestra una representación esquemática de un programa 400 de aplicación de una interfaz programable por aplicaciones entre una entidad de control y una entidad de verificación según una forma de realización. La interfaz programable por aplicaciones puede ser, por ejemplo, una interfaz 106 de verificación entre una entidad 107 de control y una entidad 103 de verificación en un sistema 100, 200 de verificación, como se ha descrito anteriormente con mayor detalle en relación con las figuras 1 y 2.

El programa informático 400 comprende un primer módulo 401 de programa, un segundo módulo 402 de programa y un tercer módulo 403 de programa. El primer módulo 401 de programa está diseñado para transmitir a la entidad 103 de verificación una identificación de un objeto virtual 109 de prueba creado por el servidor 105 de simulación, por ejemplo, una dirección del archivo 111 con datos del objeto 101 de prueba. El segundo módulo 402 de programa está diseñado para ordenar a la entidad 103 de verificación que, en lugar del ordenador 101 en cuestión, verifique el objeto virtual 109 de prueba. El tercer módulo 403 de programa está diseñado para ordenar a la entidad 103 de verificación que transmita a la entidad 107 de control resultados de la verificación del objeto virtual 303 de prueba, por ejemplo, que los almacene en una memoria de la entidad 107 de control instalada con este fin.

El programa informático 400 puede comprender además un cuarto módulo de programa, que está diseñado para ordenar a la entidad 103 de verificación que escanee el objeto virtual 101 de prueba en busca de al menos uno de los datos siguientes: versiones de *software* instalado en el objeto 101 de prueba, versiones de parches instalados en el objeto 101 de prueba, estados de configuraciones del objeto 101 de prueba.

Otro aspecto de la invención comprende un producto de programa informático que puede cargarse directamente en la memoria interna de un ordenador digital y que comprende segmentos de código de *software*, con los que puede realizarse el procedimiento 300 descrito en relación con la figura 3 cuando el producto se ejecuta en un ordenador. El producto de programa informático puede estar almacenado en un medio adecuado para un ordenador y comprender

lo siguiente: medios de programa legibles por ordenador que hagan que un servidor de simulación cree un objeto virtual de prueba para la emulación de un objeto de prueba; y ordenen a una instancia de verificación que pruebe, en lugar del ordenador en cuestión, el objeto virtual de prueba creado en el servidor de simulación.

El ordenador puede ser un PC, por ejemplo, un PC de una red de ordenadores. El ordenador puede estar realizado como un chip, un ASIC, un microprocesador o un procesador de señales y estar dispuesto en una red de ordenadores, por ejemplo, en un sistema informático como se describe en una de las figuras 1 y 2.

Es evidente que las características de las distintas formas de realización descritas a modo de ejemplo en la presente memoria pueden combinarse entre sí, excepto cuando se indique específicamente otra cosa. Como se describe en la descripción y los dibujos, los distintos elementos que se han representado como conectados no tienen que estar necesariamente conectados directamente entre sí; entre los elementos conectados pueden estar previstos elementos intermedios. Además, es evidente que algunas formas de realización de la invención pueden estar implementadas en circuitos individuales, circuitos parcialmente integrados o circuitos completamente integrados o medios de programación. La expresión “por ejemplo” alude solamente a un ejemplo y no a lo mejor u óptimo. En la presente memoria se han ilustrado y descrito determinadas formas de realización, pero para el experto en la técnica es evidente que, en lugar de las formas de realización mostradas y descritas, pueden realizarse una gran cantidad de implementaciones alternativas y/o similares sin apartarse del concepto de la presente invención.

La invención está definida por las reivindicaciones independientes 1, 9 y 11.

Lista de símbolos de referencia

100: Sistema de verificación
101: Objeto de prueba u ordenador en cuestión de un sistema informático en cuestión
102: Interfaz de prueba
103 Entidad de verificación
104: Segunda interfaz de verificación (entre el servidor de simulación y la entidad de verificación)
105 Servidor de simulación
106: Primera interfaz de verificación (entre la entidad de control y la entidad de verificación)
107: Entidad de control
108: Interfaz de control (entre la entidad de control y el servidor de simulación)
109: Objeto virtual de prueba
111a: Datos del objeto de prueba generados en el objeto de prueba
111: Datos del objeto de prueba recibidos en el servidor de simulación
117: Terminal de trabajo
T1: Transferencia de archivos entre el objeto de prueba y el servidor de simulación
T2a: Transferencia de archivos entre el objeto de prueba y el terminal de trabajo
T2b: Transferencia de archivos entre el terminal de trabajo y el servidor de simulación
300: Procedimiento para verificar
301: 1ª etapa de procedimiento: Crear un objeto virtual de prueba
302: 2ª etapa de procedimiento: Ordenar a la entidad de verificación
400: Programa informático para interfaz programable por aplicaciones entre la entidad de control y la entidad de verificación
401: 1er módulo de programa
402: 2º módulo de programa
403: 3er módulo de programa

REIVINDICACIONES

1. Sistema (100) de verificación para verificar un ordenador (101), que constituye un objeto (101) de prueba, de un sistema informático en una red de verificación, en donde la red de verificación comprende un servidor (103) de verificación para verificar ordenadores del sistema informático, en donde el sistema (100) de verificación comprende las características siguientes:

un servidor (105) de simulación, que está configurado para emular el objeto (101) de prueba; y
una entidad (107) de control, en particular un servidor de control o un *software* de aplicación en un ordenador de control, para controlar el servidor (105) de simulación, estando la entidad (107) de control configurada para ordenar al servidor (105) de simulación que cree un objeto virtual (109) de prueba para la emulación del objeto (101) de prueba, estando la entidad (107) de control configurada además para ordenar al servidor (103) de verificación que, en lugar del ordenador (101), verifique el objeto virtual (109) de prueba creado por el servidor (105) de simulación, estando el sistema (100) de verificación configurado para recibir datos (111) del objeto (101) de prueba y emular en el servidor (105) de simulación el objeto (101) de prueba basándose en los datos (111) recibidos, estando el sistema (100) de verificación configurado para recibir los datos (111) del objeto (101) de prueba a través de una transferencia directa de archivos entre el objeto (101) de prueba y el sistema (100) de verificación,
en donde la entidad (107) de control está configurada para verificar a intervalos regulares si existen datos (111) de un nuevo objeto (101) de prueba en el servidor (105) de simulación y, en caso de existir los datos (111), ordenar al servidor (105) de simulación que cree un objeto virtual (109) de prueba para la emulación del nuevo objeto (101) de prueba y ordenar al servidor (103) de verificación que verifique el objeto virtual (109) de prueba.

2. Sistema (100) de verificación según la reivindicación 1, en donde los datos (111) del objeto (101) de prueba comprenden al menos una de las siguientes informaciones:

versiones de *software* instalado en el objeto (101) de prueba,
versiones de parches instalados en el objeto (101) de prueba,
estados de configuraciones del objeto (101) de prueba.

3. Sistema (100) de verificación según una de las reivindicaciones precedentes,
en donde la entidad (107) de control está configurada para, mediante una interfaz programable por aplicaciones, recibir datos (111) y otros parámetros para el control y ordenar al servidor (105) de simulación que cree un objeto virtual (109) de prueba para la emulación del nuevo objeto (101) de prueba y ordenar al servidor (103) de verificación que verifique el objeto virtual (109) de prueba.

4. Sistema (100) de verificación según una de las reivindicaciones precedentes,
en donde la entidad (107) de control está configurada para recibir del servidor de verificación resultados de la verificación del objeto virtual de prueba.

5. Sistema (100) de verificación según una de las reivindicaciones precedentes, que comprende:

una primera interfaz (106) de verificación entre la entidad (107) de control y el servidor (103) de verificación, comprendiendo la primera interfaz (106) de verificación una interfaz programable por aplicaciones.

6. Sistema (100) de verificación según una de las reivindicaciones precedentes, que comprende:

una segunda interfaz (104) de verificación entre el servidor (105) de simulación y el servidor (103) de verificación, estando la segunda interfaz (104) de verificación configurada para cargar en el servidor (105) de simulación *plug-ins* puestos a disposición por el servidor (103) de verificación para verificar el objeto virtual (109) de prueba.

7. Sistema (100) de verificación según la reivindicación 6,
en donde el servidor (105) de simulación está diseñado para, mediante los *plug-ins* puestos a disposición por el servidor (103) de verificación, escanear el objeto virtual (109) de prueba en busca de una determinada versión de *software*, versión de parche o estado de configuración.

8. Sistema (100) de verificación según una de las reivindicaciones precedentes,
en donde el servidor (105) de simulación dispone de datos de acceso al objeto (101) de prueba, que son válidos sólo en el objeto virtual (109) de prueba y sólo para el intervalo de tiempo de la verificación del objeto virtual (109) de prueba.

9. Procedimiento para verificar un ordenador (101), que constituye un objeto (101) de prueba, de un sistema informático en una red de verificación mediante un sistema (100) de verificación, en donde la red de verificación comprende un

servidor (103) de verificación para verificar ordenadores del sistema informático y en donde el sistema (100) de verificación comprende un servidor (105) de simulación y una entidad (107) de control, con las etapas siguientes:

crear un objeto virtual (109) de prueba para la emulación del objeto (101) de prueba en el servidor (105) de simulación;
ordenar al servidor (103) de verificación, por parte de la entidad (107) de control, que, en lugar del ordenador (101) en cuestión, verifique el objeto virtual (109) de prueba creado en el servidor (105) de simulación;
en donde el crear el objeto virtual (109) de prueba en el servidor (105) de simulación y el ordenar al servidor (103) de verificación comprenden lo siguiente:

verificar (P1) a intervalos regulares si existen datos (111) de un nuevo objeto (101) de prueba en el servidor (105) de simulación, transmitiéndose los datos (111) a través de una transferencia directa de archivos entre el objeto (101) de prueba y el sistema (100) de verificación;
en caso de existir datos (111) de un nuevo objeto (101) de prueba: ordenar al servidor (105) de simulación que cree (P2) un objeto virtual (109) de prueba basándose en los datos (111) del nuevo objeto (101) de prueba;
ordenar al servidor (103) de verificación, por parte de la entidad (107) de control, que verifique (P3) el objeto virtual (109) de prueba basado en los datos (111) del nuevo objeto (101) de prueba;
recibir (P4) resultados de la verificación del objeto virtual (109) de prueba; y
borrar (P5) el objeto virtual (109) de prueba en el servidor (105) de simulación.

10. Procedimiento según la reivindicación 9, que además comprende:

escanear el objeto virtual (109) de prueba en busca de al menos uno de los datos siguientes:
versiones de *software* instalado en el objeto (101) de prueba,
versiones de parches instalados en el objeto (101) de prueba,
estados de configuraciones del objeto (101) de prueba.

11. Programa informático para una interfaz (113) programable por aplicaciones entre la entidad (107) de control de un sistema (100) de verificación según una de las reivindicaciones 1 a 9 y un servidor (103) de verificación en una red de verificación, que comprende:

un primer módulo de programa, que está diseñado para transmitir al servidor (103) de verificación una identificación de un objeto virtual (109) de prueba creado por el servidor (105) de simulación;
un segundo módulo de programa, que está diseñado para ordenar al servidor (103) de verificación que, en lugar del ordenador (101) en cuestión, verifique el objeto virtual (109) de prueba; y
un tercer módulo de programa, que está diseñado para ordenar al servidor (103) de verificación que transmita a la entidad (107) de control resultados de la verificación del objeto virtual (109) de prueba.

12. Programa informático según la reivindicación 11, que comprende:

un cuarto módulo de programa, que está diseñado para ordenar al servidor (103) de verificación que escanee el objeto virtual (101) de prueba en busca de al menos uno de los datos siguientes:
versiones de *software* instalado en el objeto (101) de prueba,
versiones de parches instalados en el objeto (101) de prueba,
estados de configuraciones del objeto (101) de prueba.

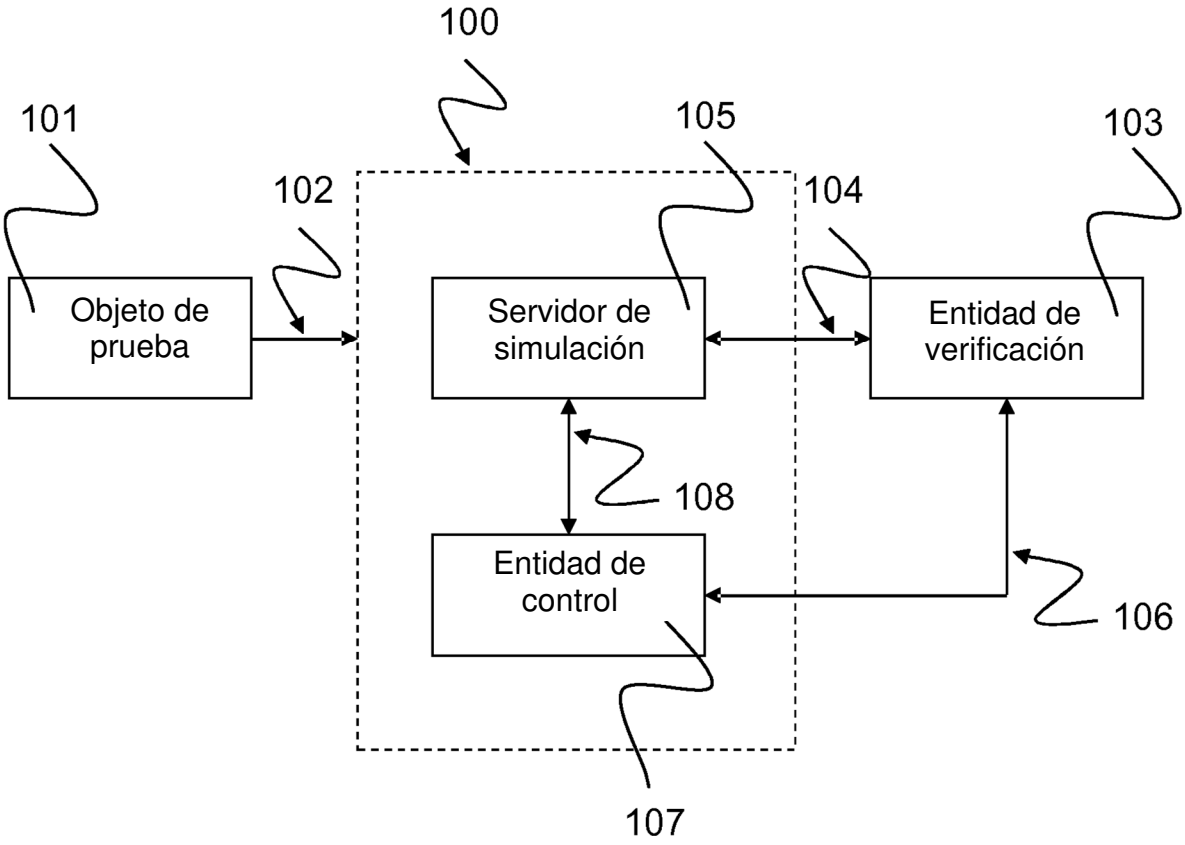
100
101
102
105
104
103
Objeto de prueba
Servidor de simulación
Entidad de verificación
108
Entidad de control
106
107

Fig. 1

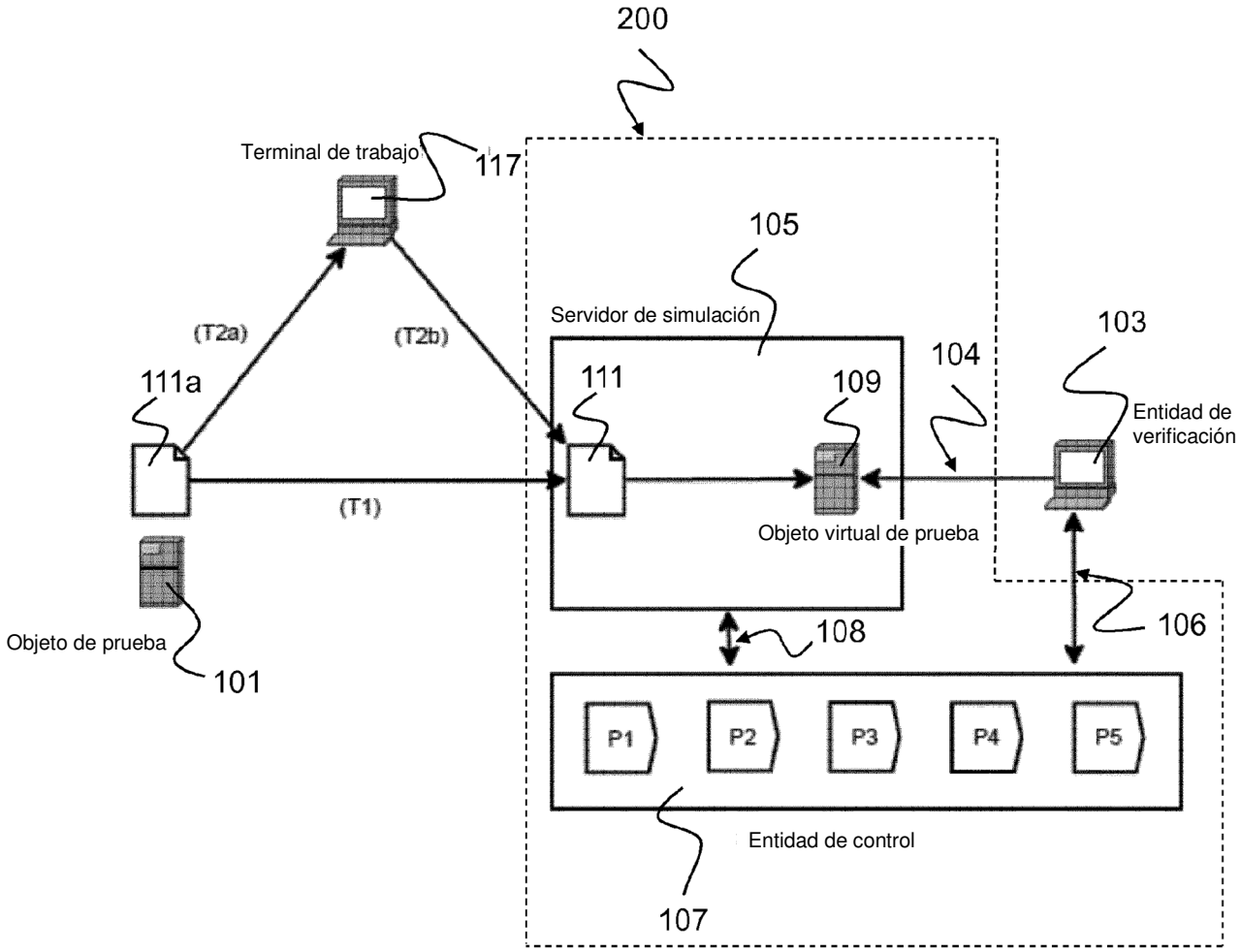
200
Terminal de trabajo
117
105
Servidor de simulación
103
104
(T2a)
(T2b)
111a
111
109
Entidad de verificación
(T1)
Objeto virtual de prueba
Objeto de prueba
101
108
106
P1
P2
P3
P4
P5
Entidad de control
107

Fig. 2

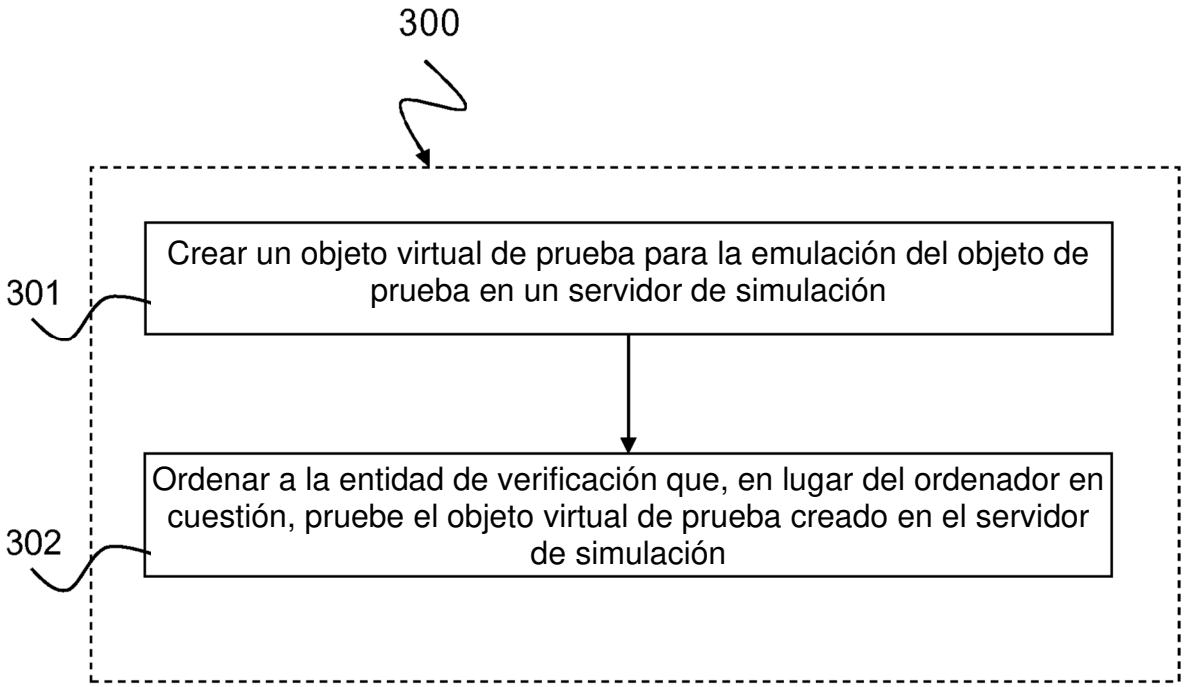
300
301
Crear un objeto virtual de prueba para la emulación del objeto de prueba en un servidor de simulación
302
Ordenar a la entidad de verificación que, en lugar del ordenador en cuestión, pruebe el objeto virtual de prueba creado en el servidor de simulación

Fig. 3

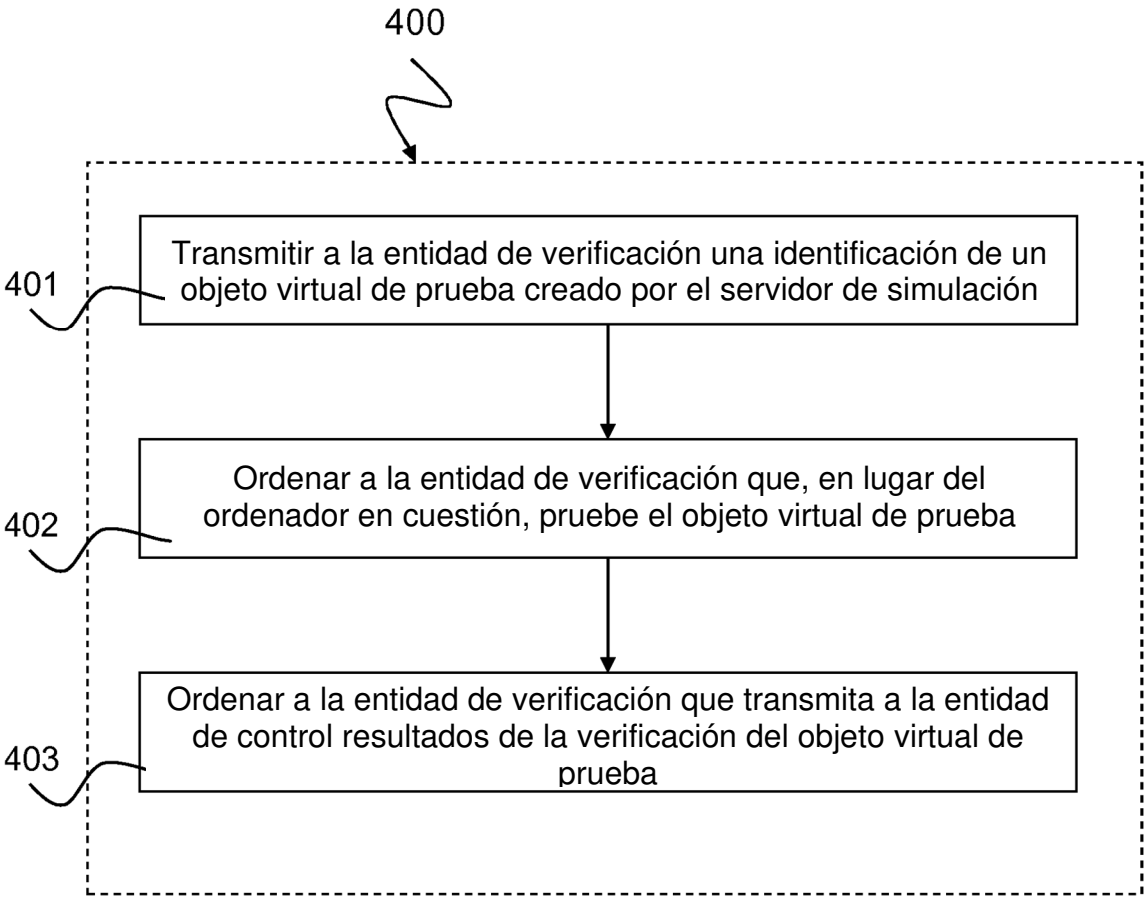
400
401
Transmitir a la entidad de verificación una identificación de un objeto virtual de prueba creado por el servidor de simulación
402
Ordenar a la entidad de verificación que, en lugar del ordenador en cuestión, pruebe el objeto virtual de prueba
403
Ordenar a la entidad de verificación que transmita a la entidad de control resultados de la verificación del objeto virtual de prueba

Fig. 4