(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 280 807**

(51) Int. Cl.:
H04L 29/06 (2006.01)
G06F 1/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Número de solicitud europea: **03767857 .0**
(86) Fecha de presentación : **29.10.2003**
(87) Número de publicación de la solicitud: **1574002**
(87) Fecha de publicación de la solicitud: **14.09.2005**

(54) Título: **Procedimiento de comunicación fiable entre dos unidades.**

(30) Prioridad: **18.12.2002 FR 02 16091**

(45) Fecha de publicación de la mención BOPI:
16.09.2007

(45) Fecha de la publicación del folleto de la patente:
16.09.2007

(73) Titular/es: **FRANCE TELECOM**
6, place d'Alleray
75015 Paris, FR

(72) Inventor/es: **De Boursetty, Benoît;**
Gruson, Manuel y
Mouton, Dimitri

(74) Agente: **Lehmann Novo, María Isabel**

DESCRIPCIÓN

Procedimiento de comunicación fiable entre dos unidades.

La presente invención se refiere a los terminales informáticos personales que permiten a los usuarios acceder a servicios en línea.

Tales terminales pueden ser especialmente teléfonos que utilizan el protocolo de aplicación inalámbrico (WAP, "wireless application protocol"), ordenadores de oficina, ordenadores portátiles o asistentes digitales personales (PDA, "personal digital assistant"). Tienen en común la característica de estar conectados a una red de datos digital, que en numerosos casos prácticos es una red que funciona según el protocolo IP ("Internet protocol"), particularmente Internet.

En estos terminales, es posible instalar diversas aplicaciones. Entre estas aplicaciones se hace con frecuencia una distinción según diversos criterios tales como su origen, el grado de fiabilidad que se les asigna por un administrador, etc., que da como resultad capacidades diferentes para ciertas aplicaciones con respecto a otras.

Por ejemplo, en los sistemas que funcionan bajo el sistema operativo llamado "Unix", los derechos de ejecución de las aplicaciones de clase "setuid root" son los derechos máximos, de nivel de administrador, mientras que los derechos de ejecución de las demás aplicaciones son simplemente los derechos de usuario que lanza la aplicación. Otro ejemplo, en los navegadores web que comprenden una máquina virtual Java, las aplicaciones, denominadas "applets", descargadas desde un sitio web dado están limitadas en cuanto a sus capacidades de acceder a la red, es decir, sólo pueden emitir peticiones del protocolo HTTP ("hypertext transfer protocol", protocolo de transferencia de hipertexto) hacia ese sitio web.

Algunos de estos derechos de ejecución de las aplicaciones son puramente locales. Es el caso, por ejemplo, del derecho de tomar el control de la pantalla de un terminal, o del derecho de tener conocimiento de todas las teclas presionadas en el teclado del terminal, incluso para otras aplicaciones.

Sin embargo, otros derechos de ejecución son observables a distancia. Es el caso de, por ejemplo, el derecho de emitir cualquier paquete IP, incluidos los paquetes IP que no se adaptasen a los formatos de los protocolos de transporte más habituales, a saber, TCP ("transmission control protocol", protocolo de control de transmisión) o UDP ("user datagram protocol", protocolo de datagrama de usuario). En los sistemas Unix, este derecho no se da a las aplicaciones que no son de la clase "setuid root". Al utilizar esta diferencia de capacidad de envío de peticiones, un observador a distancia tal como un servidor puede determinar que un paquete dado se ha emitido por una aplicación de clase "setuid root": si observa que este paquete no se adapta al formato TCP o UDP, se trata necesariamente de una aplicación de clase "setuid root", de lo contrario, puede que se trate de una aplicación sin derechos privilegiados.

En los casos de los applets en los navegadores, en los ordenadores personales, las capacidades de envío de peticiones HTTP están limitadas sólo al sitio del que se ha descargado el applet. Para cada petición HTTP recibida, un servidor web puede deducir por tanto que, o bien procede de un applet presente en el sitio o bien de otra aplicación (por ejemplo el navegador). Sin embargo, en cualquier caso, las peticiones recibidas por un servidor web no proceden de applets "extraños" presentes en otros sitios.

En el presente documento es interesante el problema de saber cómo un servidor puede recoger de forma segura la aceptación del usuario sobre una pregunta dada. La pregunta que se ha de plantear al usuario se debe presentar al usuario por medio de una aplicación en su terminal. La aplicación recoge la aceptación (o no aceptación) del usuario y después transmite una indicación correspondiente al servidor.

El servidor recibe por tanto mensajes sobre la red y los interpreta como la aceptación (o no aceptación) del usuario. Para ello se debe plantear la hipótesis de que la aplicación ha presentado adecuadamente la pregunta al usuario y ha recogido su aceptación de manera fidedigna. El servidor supone por tanto que la aplicación no es un "troyano" que habría, por ejemplo, presentado una pregunta diferente al usuario, o bien que simplemente no habría presentado la pregunta en absoluto al usuario, pero que hace como se este hubiera dado su aceptación. Para proteger al usuario de posibles programas de tipo "troyano", resulta importante garantizar esta hipótesis de fiabilidad.

Existen diversos medios para satisfacer de manera razonable esta hipótesis de fiabilidad en la aplicación.

Ciertas aplicaciones están reconocidas como "fiables". Una aplicación de este tipo es, por ejemplo, el navegador WAP. Un servidor puede confiar en un navegador WAP porque visualiza una página que plantea al usuario una pregunta y espera a que el usuario introduzca su respuesta.

En el caso de un terminal "cerrado" (por ejemplo: un Minitel), se conocen las aplicaciones presentes en el terminal y no se pueden cambiar durante la vida del terminal. Todas estas aplicaciones se reconocen como "fiables".

La apertura de un terminal hace referencia a la posibilidad que se ofrece al usuario de instalar, y a menudo descargar, nuevas aplicaciones destinadas a ejecutarse en el propio terminal. Ejemplos de terminales "abiertos", que integran esta posibilidad, son:

- los teléfonos con descarga de aplicaciones, por ejemplo de tipo Java MIDP ("Mobile Information Device Profile", Perfil de dispositivo de información móvil, Sun Microsystems, Inc.);

- los navegadores que tienen funcionalidades llamadas de "scripting" (codificación), por ejemplo de tipo WMLScript (véase "WAP WMLScript Language Specification", versión 1.1, WAP Forum, noviembre de 2001) o ECMAScript (también denominado JavaScript, véase "ECMAScript Language Specification", norma ECMA-262, 3ª edición, diciembre de 1999), o que aceptan applets;

- la mayoría de los PDA que funcionan bajo sistemas operativos PalmOS, WindowsCE, Symbian, etc.;

- los ordenadores de oficina o portátiles.

Los terminales "semiabiertos" son los terminales abiertos en los que ciertas funcionalidades no son accesibles directamente para las aplicaciones instaladas por el usuario o descargadas. Por ejemplo, en un terminal en el que la única "apertura" es ECMAScript, las aplicaciones descargadas no pueden acceder a todas las funcionalidades de la red (por ejemplo, emitir cualquier paquete IP). Estas funcionalidades pueden ser accesibles de forma indirecta o controlada. Por ejemplo, una función ECMAScript puede ordenar que se cargue una página a través de HTTP, lo que utiliza la red pero de forma controlada.

En los terminales "semiabiertos" coexisten:

- aplicaciones consideradas como "fiables", por ejemplo porque se han instalado en fábrica por el fabricante del terminal, o bien debido al hecho de la garantía que proporcionan medios tales como la firma electrónica de la aplicación, etc.; y

- otras aplicaciones que se pueden instalar en el terminal por el propio usuario, a su libre elección, pero que no acceden a los mismos derechos que las aplicaciones fiables.

Los terminales "completamente abiertos", a diferencia, son los terminales abiertos en los que todas las funcionalidades son accesibles por las aplicaciones descargadas. La noción de apertura de un terminal depende en gran medida del contexto en el que se sitúa. Por ejemplo, diferentes capas del modelo OSI (enlace/red/sesión/transporte/...) pueden tener diferentes grados de apertura.

En el presente documento son interesantes las funcionalidades observables a distancia, desde un servidor, es decir, funcionalidades de red. En este marco, el carácter "semiabierto" de un terminal implica generalmente que los derechos de ejecución observables a distancia, accesibles por las aplicaciones fiables no son accesibles por aplicaciones no fiables (por ejemplo, el derecho de emitir peticiones distintas de HTTP sobre una red IP). Esto permite a un servidor distinguir, entre las peticiones que le llegan, aquellas que proceden de aplicaciones fiables y aquellas que proceden de otras aplicaciones.

El documento "Wireless Java Security", XP2249490, da a conocer terminales informáticos personales que permiten a los usuarios acceder a servicios en línea. En estos terminales hay aplicaciones consideradas como "fiables" y "no fiables".

Los "applets" que el usuario instala a su libre elección no son necesariamente fiables para acceder a cualquier servidor. No obstante, la restricción de las peticiones de cada applet al sitio desde el que se han descargado permite a un sitio web mantener el control sobre los applets que pueden emitir peticiones hacia el mismo. Resulta por tanto razonable que el servidor suponga que las aplicaciones que presentan preguntas al usuario no son troyanos. Estas aplicaciones son por tanto "fiables", pero sólo para un sitio web.

En los terminales abiertos, hay que tener en cuenta la posibilidad de que un programa se comporte de manera engañosa con respecto al usuario (troyano). Así, nada puede garantizar a un servidor que una petición procede correctamente de un usuario y no de un programa que ha simulado la aceptación del usuario a nivel de red. Este riesgo arruina la confianza que puede tener el servidor en los datos que recibe de un cliente. La hipótesis según la cual las peticiones direccionadas al servidor reflejan las acciones el usuario no es razonable si un troyano tiene la posibilidad de enviarlas en lugar del usuario.

La respuesta clásica al riesgo del troyano es limitar las capacidades de las aplicaciones no fiables.

La limitación de la emisión de tramas desde los terminales semiabiertos se hace generalmente de forma extremadamente estricta. Sólo las aplicaciones fiables están autorizadas a emitir ciertas tramas. Esta distinción se usa para que el servidor no acepte como representativas de la aceptación del usuario tramas emitidas por aplicaciones no fiables, susceptibles de traicionar al usuario.

Resulta por lo tanto imposible que una aplicación no fiable emita tramas hacia un servidor. Es especialmente imposible que esta aplicación demuestre a este servidor la aceptación del usuario. Por ejemplo, es imposible que una aplicación no fiable proponga al usuario pagar utilizando un servidor de comercio electrónico.

Para que un "applet" que está limitado a poder emitir peticiones sólo al sitio web del que se ha descargado, la fiabilidad sólo se le concede para este servidor. Por lo tanto, es posible que este applet recoja la aceptación del usuario y transmita el resultado al sitio web del que se ha descargado. Se plantea entonces la hipótesis, razonable, de que el servidor nunca propondrá descargar aplicaciones de tipo "troyano".

Existen sistemas basados en criptografía para generar firmas electrónicas. Un ejemplo se describe en la norma "WAP WMLScript Crypto Library", WAP Forum, junio de 2001. Estos sistemas se pueden utilizar para recoger la aceptación del usuario, plantean la hipótesis de que el sistema es semiabierto, es decir, dado el caso, que las funciones de acceso a claves criptográficas no están directamente disponibles para las aplicaciones no fiables. El acceso a las claves criptográficas se gestiona por un componente de software (equipo lógico) particular, que denominaremos "componente de firma electrónica", encargado de recoger la aceptación del usuario por cuenta de la aplicación. Este componente efectúa por sí mismo la encadenación de operaciones siguientes por cuenta de aplicaciones no fiables:

- visualizar el texto que se debe firmar en la pantalla;

- esperar la confirmación del usuario;

- si se recibe una confirmación, utilizar las claves criptográficas del usuario para firmar el texto visualizado;

- en caso contrario, no firmar el texto visualizado.

Esto permite por tanto que las aplicaciones no fiables obtengan una firma electrónica de aceptación del usuario a través del componente de firma electrónica. Este procedimiento permite al servidor obtener la aceptación del usuario con respecto a un texto cualquiera.

Se ha de plantear en este caso la hipótesis de que el terminal sea completamente abierto. Si fuera posible que una aplicación no fiable accediese directamente a las funciones criptográficas, no se podría saber si la llamada a las funciones criptográficas ha estado correctamente precedida por una visualización de la totalidad del texto que se ha de firmar o si el terminal ha esperado correctamente a la aceptación del usuario antes de proceder a la firma.

Por otra parte, este procedimiento pone en práctica técnicas criptográficas que pueden resultar costosas en tiempo de ejecución, en tamaño de mensajes intercambiados sobre la red así como en consumo eléctrico (importante para los terminales portátiles). Además, la legislación sobre las técnicas criptográficas puede limitar eventualmente la posibilidad de recurrir a este procedimiento.

Por lo tanto es deseable proporcionar un comportamiento casi equivalente en términos de apertura para las aplicaciones no fiables, pero sin recurrir a la criptografía.

Un objetivo de la presente invención es permitir que una aplicación "no fiable" en medio semiabierto recoja la aceptación del usuario sobre una pregunta dada, y advertir de ello a un servidor remoto demostrándole que esto se ha realizado de manera fidedigna.

La invención propone por tanto un procedimiento de comunicación entre una primera unidad y una segunda unidad a través de una red de telecomunicación, en el que la primera unidad comprende una primera familia de aplicaciones y una segunda familia de aplicaciones que tienen capacidades de comunicación sobre la red más allá de las capacidades de comunicación de las aplicaciones de la primera familia. Según la invención, este procedimiento comprende las siguientes etapas:

/a/ un componente fiable que pertenece a la segunda familia de aplicaciones obtiene el enunciado de una pregunta que debe plantear a un usuario de la primera unidad en el marco de la ejecución de una aplicación de la primera familia;

/b/ el componente fiable presenta la pregunta a través de una interfaz de usuario y recoge una respuesta del usuario; y

/c/ para al menos un tipo de respuesta del usuario, el componente fiable transmite a la segunda unidad, a través de la red, al menos un mensaje que identifica la pregunta presentada e indica la respuesta recogida, transmitiéndose dicho mensaje en condiciones inaccesibles para las aplicaciones de la primera familia.

Otro aspecto de la presente invención se refiere a un componente de software fiable para la puesta en práctica del procedimiento anterior en el nivel de dicha primera unidad, así como a un terminal de comunicación, que incorpora tal componente de software fiable. Este componente fiable pertenece a la segunda familia de aplicaciones anteriormente mencionada e incluye instrucciones para controlar las siguientes etapas durante su ejecución en la primera unidad:

/a/ obtener el enunciado de una pregunta que se debe plantear a un usuario de la primera unidad en el marco de la ejecución de una aplicación de la primera familia;

/b/ presentar la pregunta a través de una interfaz de usuario y recoge una respuesta del usuario; y

/c/ para al menos un tipo de respuesta del usuario, transmitir a la segunda unidad, a través de la red, al menos un mensaje que identifica la pregunta presentada e indica la respuesta recogida, transmitiéndose dicho mensaje en condiciones inaccesibles para las aplicaciones de la primera familia.

Otras particularidades y ventajas de la presente invención aparecerán en la descripción a continuación de ejemplos de realización no limitativos, en referencia a los dibujos adjuntos, en los que las figuras 1 y 2 son esquemas de un sistema que pone en práctica la invención.

Se busca permitir que una unidad remota, tal como un servidor 1, obtenga de manera segura y ágil la aceptación del usuario de un terminal 2 semiabierto, con respecto a una pregunta dada. La aceptación se puede obtener por aplicaciones fiables 3, como en el caso de la navegación, pero también desde aplicaciones no fiables 4, que tienen capacidades de comunicación más restringidas (incluso inexistentes) sobre la red R de telecomunicación utilizada para dialogar con el servidor 1.

Nos situamos en este contexto en el marco de un terminal 2 que hace una distinción entre aplicaciones fiables 3 y aplicaciones no fiables 4. Esta distinción se traduce en capacidades distintas de emisión de tramas o peticiones sobre la red R. Las aplicaciones no fiables 4 están limitadas en las tramas que pueden emitir, lo que, en el esquema de la figura 1, se simboliza por una capa 5 de control que forma parte de los recursos 6 de acceso a la red con la que está equipado el terminal 2.

La capa 5 de control verifica que las tramas emitidas por las aplicaciones no fiables 4 cumplen ciertas propiedades. Si se cumplen estas propiedades, la capa de control deja pasar las tramas. En caso contrario, puede o bien no dejarlas pasar hacia la red R y avisar a la aplicación no fiable 4 que las ha emitido, o bien modificar las tramas para adecuarlas a las limitaciones de las aplicaciones no fiables. En este último caso, la trama pierde entonces su credibilidad a los ojos del servidor 1, que no la explotará.

La invención aprovecha esta capa 5 de control (cuya presencia puede ser sólo implícita y ser resultado de las propiedades del sistema operativo o, más generalmente, del entorno de ejecución de las aplicaciones en el terminal semiabierto) para impedir que una aplicación no fiable 4 emita por sí misma peticiones que probarían a un servidor la aceptación del usuario con respecto a la pregunta planteada. Por tanto, una aplicación de este tipo no puede recoger por sí misma la aceptación del usuario en una forma explicable por el servidor 1.

Se introduce así en el terminal 2, entre la aplicación no fiable, el servidor y el usuario, un componente 8 de software fiable, del que se ha asegurado previamente un comportamiento “fidedigno”. En la práctica, esta seguridad procederá a menudo del constructor del terminal semiabierto. El componente fiable 8 no se puede sustituir o modificar por una aplicación no fiable, lo que se asegura por el propio sistema semiabierto, para el que una aplicación fiable se deba mantener fiable. No existe por tanto riesgo de que el componente 8 se comporte como un troyano. Un papel primordial del componente fiable 8 es recoger la aceptación del usuario por cuenta de una o varias aplicaciones no fiables 4, por medio de una interfaz 9 de usuario del terminal.

El componente fiable 8 no está limitado en las peticiones que puede emitir, o al menos se somete a restricciones menos severas que las aplicaciones no fiables 4. En el ejemplo esquematizado en la figura 2 a continuación, no está controlado por la capa 5 de control.

Resulta interesante una aplicación 3 ó 4 que desea demostrar a un servidor remoto 1 que ha obtenido la aceptación del usuario para una pregunta dada.

Dispone inicialmente del enunciado de la pregunta así como de un dato de direccionamiento que permite ponerse en contacto con el servidor remoto, por ejemplo una indicación de tipo URL (“Uniform Resource Locator”, localizador de recursos uniforme).

Las comunicaciones de estas aplicaciones 3, 4 se someten a las siguientes reglas:

- las aplicaciones pueden efectuar comunicaciones remotas a través de los recursos 6 y de la red R, pero estas comunicaciones están limitadas por el sistema operativo semiabierto que incorpora una capa lógica 5 de control;

- cualquier servidor remoto 1, que tenga conocimiento de los límites aplicados, puede determinar si los mensajes que recibe proceden de aplicaciones fiables o no, examinando si se han aplicado los límites;

- el componente fiable 8 tiene la posibilidad de efectuar comunicaciones fuera de los límites impuestos a las aplicaciones no fiables 4, pero también dentro de estos límites si lo desea. A este respecto, se puede considerar como perteneciente a la misma familia que las aplicaciones fiables 3.

Una aplicación no fiable que desee obtener la aceptación del usuario sobre una pregunta dada y demostrar esta aceptación a un servidor remoto 1 proporciona al componente fiable 8 el enunciado de la pregunta así como la dirección del servidor. El componente fiable 8 presenta entonces la pregunta al usuario a través de la interfaz 9. El componente fiable recoge la decisión el usuario (aceptar o denegar, pudiendo interpretarse la ausencia de respuesta pasado un cierto plazo como una denegación).

Si la decisión recogida es de aceptación, el componente fiable envía al servidor una petición fuera de los límites aplicados a las aplicaciones no fiables a la dirección indicada por la aplicación 4. Esta petición contiene:

- el enunciado de la pregunta

- la respuesta del usuario.

El servidor 1 verifica, implícita o explícitamente, que la petición se ha transmitido correctamente fuera de los límites aplicados a las aplicaciones no fiables, y responde a esta petición tras la validación. La respuesta a la petición se transmite finalmente por el componente fiable 8 a la aplicación no fiable 4.

En caso de que se observe una no aceptación del usuario por el componente fiable 8, éste puede transmitir directamente a la aplicación 4 una respuesta indicando el fracaso. La respuesta negativa del usuario sólo se transmite opcionalmente al servidor 1 en este caso.

Si se fía del "componente fiable 8", el servidor remoto 1 está seguro de que las peticiones que recibe fuera de los límites corresponden adecuadamente a preguntas que se han planteado al usuario y que la elección del usuario se ha recogido correctamente. Una aplicación no fiable no puede simular este comportamiento. Por consiguiente, se descarta el riesgo de un troyano.

Si la verificación por parte del servidor 1 de la petición que se considera que indica la aceptación del usuario muestra que se ha transmitido dentro de los límites aplicados a las aplicaciones no fiables, esta petición no se interpreta como representativa de la aceptación del usuario. Esta denegación del servidor se puede notificar opcionalmente de vuelta al terminal.

Naturalmente, la pregunta presentada al usuario puede solicitar una respuesta de cualquier tipo, más compleja que "sí/no". La pregunta puede adoptar la forma, particularmente, de un formulario en el que el usuario tendría que completar varias entradas. En este caso, las diferentes entradas completadas por el usuario se pueden transmitir al servidor después de que el componente fiable 8 haya solicitado y obtenido una validación por parte del usuario.

En la descripción anterior, la aplicación no fiable 4 genera por sí misma el texto de la pregunta. Si se prefiere que el servidor 1 genere el texto de la pregunta, puede procederse por ejemplo como sigue:

- una aplicación no fiable 4 presenta al componente fiable 8 la dirección de un servidor 1 (por ejemplo un URL) y una petición apropiada para enviársela con el fin de obtener un enunciado de la pregunta que ha de plantear;

- el componente fiable 8 emite la petición a través de la red R con el fin de solicitar a este servidor 1 el enunciado de la pregunta. La petición pasa preferiblemente por la capa 5 de control para garantizar que se encuentra dentro de los límites autorizados para las aplicaciones no fiables 4;

- el servidor 1 reenvía el enunciado de la pregunta, en relación con una referencia que se recordará ulteriormente durante la transmisión de la aceptación del usuario;

- el componente fiable 8 presenta la pregunta al usuario al igual que anteriormente;

- el usuario toma su decisión;

- la decisión del usuario se recoge por el componente fiable 8;

- en caso de aceptación, el componente fiable 8 emite una petición hacia el servidor 1, esta vez fuera de los límites impuestos a las aplicaciones no fiables, incluyendo la referencia del enunciado y estipulando que el usuario ha aceptado adecuadamente (la referencia puede ser opcional, en cuyo caso el componente fiable repite el enunciado de la pregunta en la petición transmitida en este etapa; de forma general, basta con que la pregunta planteada se identifique de manera suficiente en el mensaje transmitido al servidor para indicar la aceptación del usuario);

- el servidor 1 valida la petición verificando que se ha recibido correctamente fuera de los límites impuestos a las aplicaciones no fiables, y responde a esta petición;

- la respuesta se transfiere a la aplicación que ha iniciado la solicitud.

Puesto que se ha asegurado pasar la petición procedente directamente de la aplicación no fiable 4 por la capa 5 de control, el servidor 1 sigue estando seguro de que las peticiones fuera de los límites que recibe desde el componente fiable 8 son adecuadamente resultado de una aceptación explícita del usuario.

En un modo de realización particular de la invención, el terminal dispone de una máquina virtual Java que puede corresponder al módulo 6 en la ilustración de las figuras 1 y 2. La máquina virtual permite ejecutar aplicaciones descargadas escritas en el lenguaje de programación Java desarrollado por la empresa Sun Microsystems, Inc. Todas las instrucciones del lenguaje Java se ejecutan por la máquina virtual, que recurre a las funciones de sistema después

de un determinado control. Para las aplicaciones Java, se trata adecuadamente de un entorno semiabierto puesto que no se llama a las funciones de sistema sin control.

La aplicación no fiable 4 está escrita por tanto en lenguaje Java.

En este modo de realización, los protocolos puestos en práctica por los intercambios del terminal 2 sobre la red R son protocolos HTTP (RFC 1945 ("Request for Comments"), publicada en mayo de 1996 por el IETF ("Internet Engineering Task Force")), TCP (RFC 793, IETF, septiembre de 1981) e IP (RFC 791, IETF, septiembre de 1981). El límite aplicado a las aplicaciones no fiables es que no pueden direccionar peticiones hacia los URL de tipo: "http://<servidor>/<camino>/aceptación?<serie>" en los que <servidor> es un nombre de un servidor cualquiera, <camino> es una serie de cadenas de caracteres de forma "repertorio_1/repertorio_2/.../repertorio_n" y <serie> es una cadena de caracteres cualquiera. Este límite es, por supuesto, un ejemplo, pudiendo hacer el papel cualquier otro límite. El servicio está alojado por un servidor HTTP.

El componente fiable 8 se puede implementar entonces en la máquina virtual Java por la clase UserConfirmation. Está accesible desde las aplicaciones Java 4 por una función de clase: InputStream UserConfirmation.ask (String url, String question) cuyo funcionamiento es el siguiente. Cuando una aplicación no fiable 4 recurre a la función UserConfirmation.ask(String url, String question):

- el componente fiable 8 abre una ventana o bien toma el control del terminal sobre la aplicación en curso de ejecución;

- la pregunta cuyo enunciado se da por la cadena de caracteres "question" se visualiza en la pantalla y se proponen dos elecciones al usuario, a saber, "OK" y "cancelar";

- si el usuario da su aceptación, eligiendo "OK":

  - el componente fiable 8 envía sobre la red R la petición HTTP formada por una concatenación (i) de un URL dado en parámetro ("url"), (ii) de la cadena "/aceptación?pregunta="; (iii) del enunciado de la pregunta planteada al usuario (codificada en el formato de codificación en el URL x-www-urlencoded), y de la cadena "&respuestaOK". Este comportamiento tan sólo es, por supuesto, un ejemplo que corresponde a la limitación aplicada a las peticiones procedentes de aplicaciones Java. Un servidor está seguro, gracias a esta combinación, de que las peticiones enviadas en esta fase por el componente fiable no se podrían haber enviado por las aplicaciones Java, lo que responde a la necesidad;

  - cuando el componente fiable 8 recibe a continuación la respuesta del servidor 1 (o una excepción si el servidor no está disponible), devuelve a la aplicación 4 que ha llamado, un objeto InputStream que permite a esta aplicación conocer la respuesta del servidor;

- si el usuario no da su aceptación, eligiendo "cancelar":

  - el componente fiable 8 reenvía una excepción a la aplicación 4 que ha llamado.

Para ilustrar este modo de realización particular, se considera el caso en el que el servidor gestiona un servicio de micropago que efectúa pagos en línea por cuenta del usuario mediante una simple aceptación de este último. Los pagos se cargan a una cuenta que corresponde al usuario. Cuando recibe una orden de pago, este servicio se quiere asegurar por tanto de que esta orden se confirma adecuadamente por el usuario, y no procede de un programa Java malintencionado que no habría presentado al usuario ninguna pregunta, o bien que le habría presentado una pregunta engañosa. Este servicio es, por supuesto, un ejemplo, pudiendo realizarse cualquier otro servicio que requiera la aceptación del usuario gracias a esta técnica (publicación de documentos, gestión de ficheros, mensajería, etc.).

En este ejemplo, el servicio de pago controla el sitio web "pago.com". Cuando una aplicación no fiable desea proponer un pago al usuario, recurre a una función UserConfirmation.ask dándole como parámetros:

- como URL: http://pago.com/pago

- como enunciado de la pregunta: "¿Pagar 1 € a Acme Co.?".

El componente fiable 8 toma el control del terminal 2, y pregunta al usuario "¿Pagar 1 € a Acme Co.? OK/ cancelar". Si el usuario elige el enlace "OK", el componente fiable emite la petición "http://pago.com/pago/aceptación?pregunta=¿Pagar+1€+a+Acme+Co.?&respuesta=OK" y transmite la respuesta del servidor a la aplicación 4 que ha llamado, devolviéndole el turno.

Si el usuario elige el enlace "cancelar", el componente fiable 8 no emite ninguna petición y devuelve una excepción a la aplicación 4 que ha llamado.

Si una aplicación 4 intenta solicitar directamente la página "http://pago.com/pago/aceptación?pregunta=¿Pagar+1€+a+Acme+Co.?&respuesta=OK", esta petición se deniega por la limitación aplicada a las aplicaciones no fiables.

Como otra ilustración del procedimiento según la invención, se considera el caso en el que el servidor gestiona un servicio de comercio electrónico. En el marco de un servicio de este tipo, se pide al cliente que rellene un formulario de pedido. Este formulario se debe enviar según el método HTTP POST a la dirección http://servicio.com/pedido.

El componente fiable se puede implementar entonces en la máquina virtual Java. Es accesible por las aplicaciones Java mediante una función "UserConfirmation.askForm(String url, byte[]formulario)".

Cuando una aplicación Java 4 recurre a esta función, el componente fiable 8:

- visualiza en la pantalla el formulario contenido en la tabla "formulario" que se ha pasado como parámetro de la función. Este formulario está, por ejemplo, en un formato XML;

- deja que el usuario rellene los campos del formulario y le pide que lo valide eligiendo "OK" o "cancelar" al final del formulario;

- envía una petición HTTP POST cuando el usuario valida el formulario, al URL "url+/aceptación?", conteniendo esta petición el formulario que se ha presentado al usuario así como los datos introducidos por el usuario en los diferentes campos.

Si una aplicación Java 4 no fiable intenta acceder a la dirección "url+/aceptación?", la petición ser denegará por la capa de control.

Por otra parte, una aplicación podría intentar inducir un error en el usuario haciéndole rellenar un formulario que comprende las mismas entradas que el formulario auténtico, pero con una redacción diferente. Este ataque se frustra igualmente por el hecho de que el formulario se transmite al servidor 1 por el componente fiable 8. De este modo, el servidor 1 puede verificar en efecto que el formulario rellenado por el usuario es adecuadamente un formulario legítimo.

Para aclarar la descripción se ha tomado un ejemplo sencillo de limitación impuesta a las aplicaciones no fiables, a saber, ciertos URL no son accesibles, lo que se controla en el momento de la emisión de una petición. No obstante, sería aceptable cualquier otra limitación.

En particular se puede utilizar un bloqueo completo de cualquier acceso a la red R por las aplicaciones no fiables 4, un bloqueo selectivo que autorice sólo las peticiones hacia el sitio web de origen de una aplicación descargada, etc.

La limitación también se puede referir a una marcación específica asociada o bien a las aplicaciones no fiables 4, o bien a las aplicaciones fiables 3. Cada petición procedente de una aplicación no fiable 4, emitida sobre la red R con destino al servidor 1, está limitada por tanto por la capa 5 de control:

/1/ ya sea para incluir una marcación asociada a la familia de las aplicaciones no fiables,

/2/ ya sea para no incluir una marcación asociada a la familia de las aplicaciones fiables, estando esta marcación incluida por tanto en al menos algunas de las peticiones emitidas sobre la red R y procedentes de las aplicaciones fiables.

En el caso /1/, el componente fiable 8 no fija la marcación en las peticiones emitidas para indicar la aceptación del usuario, lo que garantiza al servidor 1 que esta aceptación procede adecuadamente del usuario. El componente fiable 8 puede, por el contrario, marcar la petición emitida sobre la red R para obtener el enunciado de la pregunta que se ha de plantear, en el caso en que este enunciado no se proporcione directamente por la aplicación 4.

A la inversa, en el caso /2/, el componente fiable 8 fija la marcación en las peticiones emitidas para indicar la aceptación del usuario, y dado el caso, no marca la petición emitida sobre la red R para obtener el enunciado de la pregunta que se ha de plantear.

En el ejemplo en el que el componente fiable 8 forma parte de una máquina virtual Java 6, la marcación del caso /1/ consiste por ejemplo en que el campo de cabecera "User-Agent" de las peticiones HTTP (véase la sección 10.15 de la RFC 1945 anteriormente mencionada) contiene una cadena específica tal como "Aplicación no fiable: VM Java 1.2" que indica por su presencia que la petición no procede de una aplicación fiable. Una mención inversa se puede prever en el caso /2/.

REIVINDICACIONES

1. Procedimiento de comunicación entre una primera unidad (2) y una segunda unidad (1) a través de una red (R) de telecomunicación, en el que la primera unidad comprende una primera familia de aplicaciones (4) y una segunda familia de aplicaciones (3) que tienen capacidades de comunicación sobre la red más allá de las capacidades de comunicación de la primera familia, estando el procedimiento **caracterizado** por las etapas siguientes:

/a/ un componente fiable (8) que pertenece a la segunda familia de aplicaciones obtiene el enunciado de una pregunta que debe plantear a un usuario de la primera unidad en el marco de la ejecución de una aplicación (4) de la primera familia;

/b/ el componente fiable presenta la pregunta a través de una interfaz (9) de usuario y recoge una respuesta del usuario; y

/c/ para al menos un tipo de respuesta del usuario, el componente fiable transmite a la segunda unidad, a través de la red, al menos un mensaje que identifica la pregunta presentada e indica la respuesta recogida, transmitiéndose dicho mensaje en condiciones inaccesibles para las aplicaciones de la primera familia.

2. Procedimiento según la reivindicación 1, en el que la pregunta presentada se identifica en el mensaje de la etapa /c/ incluyendo el enunciado de la pregunta en dicho mensaje.

3. Procedimiento según la reivindicación 1 ó 2, en el que para al menos otro tipo de respuesta que traduce una denegación del usuario con respecto a la pregunta presentada, el componente fiable (8) indica la denegación a dicha aplicación (4) de la primera familia.

4. Procedimiento según la reivindicación 3, en el que para el tipo de respuesta que traduce una denegación del usuario con respecto a la pregunta presentada, el componente fiable (8) no transmite a la segunda unidad (1) el mensaje de la etapa /c/.

5. Procedimiento según una cualquiera de las reivindicaciones precedentes, en el que la segunda unidad (1) valida la respuesta del usuario con la recepción del mensaje transmitido en la etapa /c/ asegurándose de que se ha transmitido adecuadamente en condiciones inaccesibles por las aplicaciones de la primera familia.

6. Procedimiento según la reivindicación 5, en el que después de la validación de la respuesta del usuario, la segunda unidad (1) devuelve un mensaje de respuesta al componente fiable (8) a través de la red (R).

7. Procedimiento según la reivindicación 6, en el que el componente fiable (8) indica a dicha aplicación (4) de la primera familia el contenido del mensaje de respuesta recibido desde la segunda unidad (1).

8. Procedimiento según una cualquiera de las reivindicaciones anteriores, en el que el enunciado de la pregunta se indica directamente al componente fiable (8) en la etapa /a/ por dicha aplicación (4) de la primera familia.

9. Procedimiento según la reivindicación 8, en el que dicha aplicación (4) de la primera familia indica una dirección de la segunda unidad (1) con el enunciado de la pregunta en la etapa /a/.

10. Procedimiento según una cualquiera de las reivindicaciones 1 a 7, en el que la etapa /a/ comprende las siguientes etapas secundarias:

/a1/ dicha aplicación (4) de la primera familia indica al componente fiable (8) una dirección de la segunda unidad (1) así como una petición de presentación para obtener el enunciado de la pregunta por parte de la segunda unidad;

/a2/ el componente fiable emite la petición a la dirección indicada, a través de la red (R);

/a3/ el componente fiable recupera el enunciado de la pregunta en una respuesta a la petición devuelta por la segunda unidad a través de la red.

11. Procedimiento según la reivindicación 10, en el que la petición se emite por el componente fiable (8) en la etapa secundaria /a2/ en condiciones accesibles para las aplicaciones de la primera familia.

12. Procedimiento según la reivindicación 10 u 11, en el que la respuesta a la petición devuelta por la segunda unidad (1) incluye además una referencia que el componente fiable (8) memoriza e inserta después en el mensaje transmitido en la etapa /c/ para identificar la pregunta presentada.

13. Procedimiento según una cualquiera de las reivindicaciones anteriores, en el que dicha aplicación (4) de la primera familia es un programa escrito en lenguaje Java y el componente fiable (8) está incorporado en una máquina virtual Java (6) con la que está equipada la primera unidad (2).

14. Procedimiento según una cualquiera de las reivindicaciones anteriores, en el que las aplicaciones (3) de la segunda familia tienen la capacidad de acceder, a través de la red (R), a al menos un URL asociado a la segunda unidad (1) e inaccesible por las aplicaciones (4) de la primera familia.

15. Procedimiento según una cualquiera de las reivindicaciones 1 a 13, en el que las aplicaciones (4) de la primera familia no pueden acceder a la red (R).

16. Procedimiento según una cualquiera de las reivindicaciones 1 a 13, en el que las aplicaciones (4) de la primera familia tienen la capacidad, en un protocolo de transferencia determinado, de acceder sólo a un sitio remoto que no comprende la segunda unidad (1).

17. Procedimiento según una cualquiera de las reivindicaciones 1 a 13, en el que cada petición procedente de una aplicación (4) de la segunda familia, emitida sobre la red (R) con destino a la segunda unidad (1), se limita a incluir una marcación asociada a la segunda familia de aplicaciones (3).

18. Procedimiento según una cualquiera de las reivindicaciones 1 a 13, en el que cada petición procedente de una aplicación (4) de la segunda familia, emitida sobre la red (R) con destino a la segunda unidad (1), se limita a no incluir una marcación asociada a la primera familia, incluyéndose dicha marcación en al menos algunas de las peticiones emitidas sobre la red y procedentes de aplicaciones (3) de la primera familia.

19. Procedimiento según la reivindicación 17 ó 18, en el que las peticiones comprenden peticiones HTTP, y la marcación se inserta en las cabeceras de las peticiones HTTP.

20. Componente de software fiable para una primera unidad (2) que se puede comunicar con una segunda unidad (1) a través de una red (R) de telecomunicación, comprendiendo la primera unidad una primera familia de aplicaciones (4) y una segunda familia de aplicaciones (3) que tienen capacidades de comunicación sobre la red más allá de las capacidades de comunicación de las aplicaciones de la primera familia, perteneciendo el componente fiable (8) a la segunda familia de aplicaciones e incluyendo instrucciones para controlar las etapas de un procedimiento según una cualquiera de las reivindicaciones 1 a 19, durante una ejecución del componente en la primera unidad.

21. Terminal de comunicación, que incorpora un componente de software fiable según la reivindicación 20, para comunicarse con una unidad (1) remota a través de una red (R) de telecomunicación.

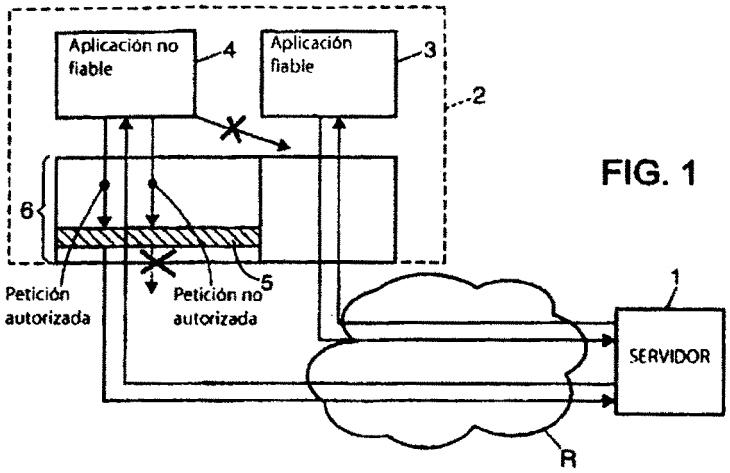
Aplicación no fiable
4
Aplicación fiable
3
2
6
Petición autorizada
Petición no autorizada
5
1
SERVIDOR
R

FIG. 1

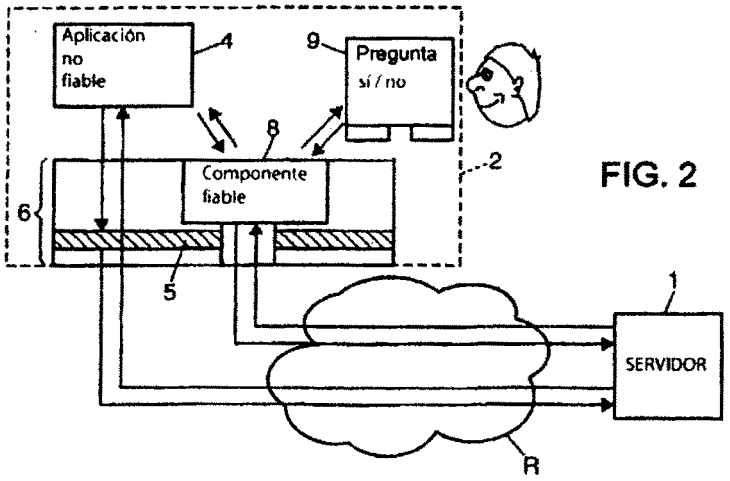
Aplicación no fiable
4
9
Pregunta sí / no
8
Componente fiable
2
6
5
1
SERVIDOR
R

FIG. 2