(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 343 623**

(51) Int. CI.:

G06F 21/00 (2006.01)

(12) PATENTE EUROPEA LIMITADA EN ESPAÑA B5

(96) Fecha de presentación y número de la solicitud europea: **28.05.2003 E 03727702 (7)**

(97) Fecha y número de publicación de la solicitud europea: **09.03.2005 EP 1512058**

(54) Título: **Dispositivo inalámbrico móvil seguro**

(30) Prioridad:

28.05.2002 GB 0212314

Fecha de resolución de limitación de la patente:
24.09.2020

(45) Fecha de publicación y mención en BOPI de la patente europea limitada en España:

01.10.2020

(73) Titular/es:

NOKIA TECHNOLOGIES OY (100.0%)
Karaportti 3
FI-02610 ESPOO, FI

(72) Inventor/es:

DIVE-RECLUS, Corinne;
HARRIS, Jonathan y
MAY, Dennis

(74) Agente/Representante:

GONZÁLEZ PECES, Gustavo Adolfo

ES 2 343 623 B5

DESCRIPCIÓN

Dispositivo inalámbrico móvil seguro

Campo de la invención

La presente invención versa acerca de un dispositivo inalámbrico móvil seguro; en particular, versa acerca de un nuevo enfoque de la seguridad de plataformas para los dispositivos inalámbricos móviles que protege recursos clave del sistema operativo contra un acceso dañino.

Descripción de la técnica anterior

La seguridad de plataformas abarca la filosofía, la arquitectura y la implementación de mecanismos de defensa de las plataformas contra código malicioso o mal escrito. Estos mecanismos de defensa evitan que tal código cause daño. Generalmente, el código malicioso tiene dos componentes: un mecanismo de carga útil que causa el daño y un mecanismo de propagación que lo ayuda a extenderse. Suelen clasificarse como sigue:

Caballo troyano: se presenta como una aplicación legítima que parece benigna y atractiva para el usuario.
Gusano: puede replicarse y extenderse sin acción manual adicional por parte ni de los perpetradores ni de los usuarios.
Virus: se infiltra en los programas legítimos y altera los datos o los destruye.

Las amenazas a la seguridad abarcan (a) una brecha potencial de la confidencialidad, de la integridad o de la disponibilidad de los servicios o de los datos en la cadena de valores y la integridad de los servicios y (b) la puesta en peligro de la función del servicio. Las amenazas se clasifican en las categorías siguientes:

1. Amenazas a la confidencialidad y la integridad de los datos: Ejemplos: Obtener la contraseña del usuario; corromper ficheros.
2. Amenazas a la confidencialidad y la integridad de los servicios. Ejemplos: Usar ancho de banda de abono a la red telefónica sin pagarlo; repudio de transacciones con el proveedor de servicios de red
3. Amenazas a la disponibilidad de un servicio (también denominadas denegación de servicio). Ejemplos: Impedir que el usuario envíe un mensaje de texto; impedir que el usuario acepte una llamada de teléfono.

Por ende, los dispositivos inalámbricos móviles ofrecen retos muy considerables para quien diseña una arquitectura de seguridad. Típicamente, un enfoque convencional actúa determinando en primer lugar si el código (por ejemplo, una aplicación) es lo suficientemente digna de confianza como para ser instalada y después, proteger subsiguientemente la plataforma contra su conducta errática una vez que se ha instalado. En términos de la determinación de la confianza, las aplicaciones de terceros pueden catalogarse según el grado de confianza asociado con sus autores. Por ello, se conoce el permiso para que las aplicaciones se instalen automáticamente en un dispositivo únicamente si pueden presentar un certificado digital válido emitido por una Autoridad Certificadora (AC) aceptable.

Sin embargo, los certificados digitales solo se están empezando a extender ahora; antes de su adopción, un mecanismo primario para la seguridad de las plataformas debía conceder privilegios a un usuario en vez de a una aplicación. En el momento de la ejecución, se le conceden al código que ejecuta el usuario todos los privilegios de este. En el momento de la instalación, los programas sensibles únicamente podían ser instalados por usuarios designados. Esto significaba que el código potencialmente dañino podía ser examinado por un administrador experto antes de su instalación. Muchos sistemas operativos bien conocidos adoptan este enfoque a la seguridad de la plataforma; por ejemplo, los sistemas operativos basados en Unix.

Sin embargo, la seguridad de plataformas basada en dar a los distintos usuarios diferentes privilegios de acceso no es relevante para los dispositivos inalámbricos móviles por la sencilla razón de que esta categoría de producto es para un solo usuario. Es un dispositivo personal, típicamente un “teléfono inteligente”, un teléfono móvil mejorado, una PDA u otro dispositivo informático personal portátil. Además, es muy probable que el usuario único no sea experto en informática y, por ello, que no pueda evaluar de manera fiable el riesgo de instalar y ejecutar código. Con los dispositivos inalámbricos móviles, las amenazas clave que un modelo de seguridad de plataformas intenta abordar son el acceso no autorizado a los datos del usuario y a los servicios del sistema, en particular a la pila de operaciones del teléfono. La pila de operaciones del teléfono es especialmente importante, porque controla una conexión permanente de datos con la red telefónica; el comercio móvil, los servicios bancarios móviles, etc., se llevarán a cabo usando la pila de operaciones del teléfono (como acaba de hacerse notar). Perder el control de la misma a código malicioso o mal escrito expone al propietario del dispositivo potencialmente a un daño financiero muy significativo.

Hasta la fecha, no ha habido propuestas efectivas para la seguridad de plataforma para los dispositivos inalámbricos móviles.

El documento EP 0 813 133 A2 da a conocer un mecanismo para usar contenido firmado. Se suministra el contenido a una máquina cliente en la que es capaz de usar los recursos informáticos de la máquina. El contenido pueden ser miniaplicaciones Java. El contenido incluye una firma que describe las credenciales de seguridad del creador y los

Raíz (obligatorias) "Pleno acceso a todos los ficheros: Puede modificar las capacidades asociadas con los ejecutables"

requisitos de recursos del contenido. Un gestor de seguridad usa la información de la firma para confeccionar capacidades que conceden y regulan al acceso a diferentes subconjuntos de los recursos informáticos.

Resumen de la presente invención

En un primer aspecto de la presente invención, se presenta un dispositivo inalámbrico móvil para un único usuario en el que se instala código ejecutable nativo, incluyendo el dispositivo: una pluralidad de recursos protegidos; una pluralidad de servidores; y

una base informática de confianza que tiene un núcleo; en el que el acceso a dicho recurso protegido es proporcionado por un correspondiente servidor;
al código ejecutable nativo se le asigna un conjunto de capacidades que definen un/los recurso(s) protegido(s) en el dispositivo a los que puede acceder el código ejecutable nativo;
dichos servidores correspondientes están dispuestos para que vigilen el acceso a dichos recursos protegidos en base a las capacidades asignadas al código ejecutable nativo;
las capacidades están almacenadas en una ubicación que es accesible únicamente a la base informática de confianza; y
el núcleo está dispuesto, para cada comunicación cliente-servidor, para pasar las capacidades del cliente a dicho servidor.

En un segundo aspecto, se presenta un procedimiento para habilitar que un código ejecutable nativo, instalado en un dispositivo inalámbrico móvil para un único usuario, acceda a recursos protegidos en el dispositivo, en el que:

el dispositivo incluye: una pluralidad de dichos recursos protegidos; una pluralidad de servidores; y una base informática de confianza que tiene un núcleo;
el acceso a cada uno de dichos recursos protegidos es proporcionado por un correspondiente servidor;
al código ejecutable nativo se le asigna un conjunto de capacidades que definen un/los recurso(s) protegido(s) en el dispositivo a los que puede acceder el código ejecutable nativo; y
las capacidades se almacenan en una ubicación que es accesible únicamente a la base informática de confianza;
comprendiendo el procedimiento las etapas de:

(a) vigilar el acceso a dichos recursos protegidos en dichos servidores correspondientes en base a las capacidades asignadas al código ejecutable nativo; y
(b) para cada comunicación cliente-servidor, el núcleo pasa las capacidades del cliente a dicho servidor;
(c) comprobar, en dichos servidores correspondientes, las capacidades del cliente en cada llamada relevante de un servidor realizada por un cliente.

Por lo tanto, la presente invención toma la idea de las capacidades (conocida en el contexto de definir las capacidades o privilegios de acceso de diferentes usuarios en un sistema multiusuario) y la aplica a definir las capacidades o los privilegios de acceso de diferente código ejecutable nativo para dispositivos inalámbricos móviles seguros de un solo usuario.

Puede concebirse una implementación de la presente invención como una protección de cortafuegos de servidores clave del sistema operativo mediante el uso de un control de acceso basado en capacidades tal como se aplica a código ejecutable nativo. Cada capacidad puede conceder el acceso a una API (o a una gama de API), a un fichero específicos o a cualquier/cualesquiera otro(s) recurso(s) aplicable(s). Cada ejecutable (por ejemplo, programas (EXE), o bibliotecas compartidas estáticas o dinámicas (DLL)) contiene las capacidades que se le han concedido en el momento de la instalación. Cada ejecutable está almacenado de manera permanente en una ubicación accesible únicamente a la Base Informática de Confianza (núcleo, cargador, servidor de ficheros e instalador de programas; véase la Descripción detallada, sección 1.1) para mantener la integridad de las capacidades asignadas al ejecutable. Una vez que se invoca un ejecutable, el cargador carga el ejecutable, ya sea para crear un nuevo proceso o para cargar el código de la biblioteca en un proceso existente. Cuando se crea un nuevo proceso, las capacidades de este proceso son iguales a las capacidades concedidas al programa. En base a este conjunto de capacidades, el cargador controla qué bibliotecas pueden cargarse en el proceso n. Para cada comunicación cliente-servidor, el núcleo pasa las capacidades del cliente al servidor. Por lo tanto, el servidor puede confiar plenamente en las capacidades del cliente, dado que no son pasadas/aprobadas por él, sino por una parte de un proceso de la Base Informática de Confianza, por ejemplo el núcleo. El servidor puede decidir o procesar la llamada o rechazarla.
Preferentemente, el modelo de capacidades está limitado de forma deliberada a un número pequeño de capacidad.

Las capacidades del sistema son obligatorias y controlan el acceso al núcleo del sistema operativo, al sistema de ficheros y a los datos del sistema. Garantizan la integridad del "Entorno Informático de Confianza" (o EIC, véase la Descripción detallada en 1.2). La forma en que son controladas no puede ser modificada por el usuario del dispositivo, y nunca están expuestas a ellos. Otros tipos de capacidad son discrecionales, dado que son lo suficientemente significativas para los usuarios medios como para dejarlos decidir si conceder o no algunas de ellas al código que han instalado. Estas capacidades se catalogan como sigue:

Usadas únicamente por la Base Informática de Confianza.

Capacidades del sistema (obligatorias)

Como ejemplos, podemos definir:
EscribirDatosSistema permite a un proceso modificar los datos del sistema de configuración.
ComDS concede acceso a todos controladores de los dispositivos de comunicaciones y de tarjetas de Ethernet.

Capacidades expuestas al usuario (discrecionales)

Como ejemplos, podemos definir:

RedTelefónica. “Puede acceder a servicios de la red telefónica (y potencialmente gastar dinero del usuario)” LeerDatosUsuario. “Puede leer la información privada del usuario del dispositivo”.

Breve descripción de los dibujos

La presente invención será descrita con referencia a los dibujos adjuntos, en los que

la **Figura 1** muestra los procesos del momento de ejecución, los anillos de capacidades del sistema y agrupaciones de capacidades visibles para el usuario para una implementación; y

la **Figura 2** ilustra un posible procedimiento de instalación.

Descripción detallada

La presente invención será descrita con referencia a la arquitectura de seguridad del sistema operativo orientado a objetos SO Symbian, diseñado para dispositivos inalámbricos de un solo usuario. El sistema operativo Symbian ha sido desarrollado para dispositivos inalámbricos móviles por Symbian Ltd, de Londres, Reino Unido.

El esquema básico de la arquitectura de seguridad del SO Symbian es análogo a las defensas de un castillo medieval. De forma similar, emplea capas simples y escalonadas de seguridad por encima y más allá del perímetro de la instalación. Las amenazas clave que el modelo intenta abordar son aquellas relacionadas con el acceso no autorizado a datos del usuario y a servicios del sistema, en particular a la pila de operaciones del teléfono. La pila de operaciones del teléfono es especialmente importante en el contexto de un teléfono inteligente, porque estará controlando una conexión de datos permanente con la red telefónica. Hay dos controladores clave de diseño que están tras el modelo:

- Una **protección de cortafuegos** de los recursos clave del sistema mediante el uso de un **control de accesos basado en capacidades.**
- Una **partición de los datos**, lo que crea una parte protegida del sistema de ficheros a la que los programas estándar no son capaces de acceder.

El principal concepto del modelo de capacidades descrito más abajo es controlar lo que puede hacer un proceso en vez de lo que puede hacer un usuario. Este enfoque es muy diferente del de sistemas bien conocidos, como Windows NT y Unix. Las razones principales son:

- La propia naturaleza del SO Symbian es que sea monousuario.
- El SO Symbian proporciona servicios por medio de procesos servidores independientes. Siempre están en funcionamiento y no están vinculados a una sesión de usuario. Mientras se suministre energía, el SO Symbian siempre está activo, aunque no haya un usuario conectado.
- El SO Symbian está concebido para que sea utilizado en dispositivos usados por el gran público sin conocimiento tecnológico. Cuando instala programas, el usuario puede no tener la aptitud de decidir qué permisos conceder a una aplicación. Además, con dispositivos que están siempre conectados, las consecuencias de una decisión errónea o malévola puede tener un impacto mucho mayor en un dominio que en el propio dispositivo.

Un detalle notable de esta invención es que el uso de la criptografía ha sido excluido del núcleo y del cargador de forma deliberada: los ejecutables no van firmados, las capacidades no se almacenan cifradas ni firmadas. Ello ha sido posible porque la partición de los datos proporciona una zona de almacenamiento segura para los ejecutables. Permite que el sistema trate los ejecutables de la ROM (dispositivo incorporado) de la misma manera que el código de la RAM (instalada por el usuario) sin poner en peligro ni la seguridad ni el rendimiento. Por lo tanto, todo uso indebido o vulnerabilidad del código del fabricante del dispositivo está limitado a las capacidades asignadas a este código.

1 ***Plataforma informática de confianza***

1.1. ***Base informática de confianza***

Una base informática de confianza (BIC) es un requisito arquitectónico básico para una seguridad robusta de una plataforma. La base informática de confianza consiste en varios elementos arquitectónicos que no pueden ser subvertidos y que garantizan la integridad del dispositivo. Es importante mantener esta base en el menor tamaño posible y aplicar el principio de privilegio mínimo para garantizar que a los servidores del sistema y a las aplicaciones no hay que darles privilegios que no necesiten para funcionar. En los dispositivos cerrados, la BIC consiste en el núcleo, el cargador y el servidor de ficheros; en los dispositivos abiertos, también se requiere el instalador de programas. Todos estos procesos son fidedignos en la totalidad del sistema y, por lo tanto, gozan de pleno acceso al dispositivo. Este núcleo de confianza se ejecutaría con una capacidad de "raíz" no disponible a otro código de la plataforma (véase la sección 2.1).

Hay otro elemento importante para mantener la integridad de la base informática de confianza que está fuera del ámbito de la presente invención: concretamente, los componentes físicos. En particular, con dispositivos que mantienen la funcionalidad de la base informática de confianza en ROM *flash*, es necesario proporcionar un cargador de arranque seguro para garantizar que no es posible subvertir la base informática de confianza con una imagen maliciosa de la ROM.

1.2. ***Entorno informático de confianza***

Más allá del núcleo, se concederían capacidades ortogonales restringidas del sistema a otros componentes del sistema, y constituirían el Entorno Informático de Confianza (EIC); incluirían servidores del sistema, como los servidores de teléfono y de ventanas... Por ejemplo, al servidor de ventanas no se le concedería la capacidad de acceso a la pila de operaciones del teléfono, y al servidor de teléfono no se le concedería la capacidad del acceso directo a los eventos del teclado. Se recomienda encarecidamente dar el menor número posible de capacidades a un componente de programación para limitar el daño potencial provocado por cualquier uso indebido de estos privilegios.

La BIC garantiza la integridad de todo el sistema, dado que cada elemento del EIC garantiza la integridad de un servicio. El EIC no puede existir sin una BIC, pero la BIC puede existir por sí sola para garantizar un "cajón de arena" seguro para cada proceso.

2 ***Capacidades de los procesos***

Se puede interpretar que una capacidad es un testigo de acceso que se corresponde a un permiso para emprender una acción sensible. El propósito del modelo de capacidades es controlar el acceso a los recursos sensibles del sistema. El recurso más importante que requiere control de acceso es el propio ejecutable del núcleo, y se requiere una capacidad de sistema (véase la sección 2.1) en un cliente para acceder a cierta funcionalidad mediante la API del núcleo. Todos los demás recursos residen en servidores del lado del usuario a los que se accede por medio de una CEP [Comunicación entre procesos]. Se definiría un conjunto pequeño de capacidades básicas para vigilar ciertas acciones específicas del cliente en los servidores. Por ejemplo, la posición de una capacidad de hacer llamadas permitiría que un cliente usara el servidor telefónico. Sería responsabilidad del correspondiente servidor vigilar el acceso del cliente a los recursos representados por la capacidad. Las capacidades también estarían asociadas con cada biblioteca (DLL) y cada programa (EXE) y el cargador las combinaría en el momento de la ejecución para producir capacidades netas de proceso que serían mantenidas por el núcleo. Para los dispositivos abiertos, a los programas de terceros se les asignarían capacidades ya sea durante la instalación de los programas en base al certificado usado para firmar sus paquetes de instalación, o con posterioridad a la instalación de los programas por el usuario, tal como se detalla en la sección 3. La vigilancia de las capacidades se gestionaría entre el cargador, el núcleo y los servidores afectados, pero contaría con la mediación del núcleo por medio del mecanismo de la CEP.

Las características clave del modelo de las capacidades de los procesos son:

- Se centra fundamentalmente en torno a servidores del sistema y en interacciones CEP cliente-servidor entre estas entidades.
- Las capacidades están asociadas con procesos y no hilos. Los hilos del mismo proceso comparten el mismo espacio de direcciones y los mismos permisos de acceso a la memoria. Esto significa que cualesquiera datos que esté usando un hilo pueden ser leídos y modificados por todos los demás hilos en el mismo proceso.
- La vigilancia de las capacidades es gestionada por el cargador y el núcleo y mediante la vigilancia de las capacidades en los servidores objetivo. En esta está implicado el mecanismo CEP del núcleo.
- Cuando el código no se está ejecutando, las capacidades se almacenan dentro de las bibliotecas y los programas. Las capacidades almacenadas en las bibliotecas y los programas no son modificables, dado que se almacenarían durante la instalación en una ubicación que es accesible únicamente por la base informática de confianza.
- No todos los servidores tendrían que manipular las capacidades del cliente. Los servidores serían responsables de interpretar las capacidades a su antojo.
- La única criptografía implicada en este modelo sería en la etapa de instalación de los programas,

en la que los certificados serían verificados con un certificado raíz adecuado (refiérase a la subsección 3).

2.1. Capacidades del sistema: Protección de la integridad del dispositivo

Raíz. "Pleno acceso a todos los ficheros*: Puede modificar las capacidades asociadas con los ejecutables"*
Capacidad "raíz". Usada únicamente por la base informática de confianza. Da pleno acceso a todos los ficheros del dispositivo.

Capacidades del sistema

Algunos servidores del sistema requieren algún acceso específico a la base informática de confianza. Debido a la implementación orientada a objetos del SO Symbian, el tipo de recursos requeridos por un servidor de sistema es exclusivo al mismo la mayor parte del tiempo. Por lo tanto, a un servidor del sistema se le concedería alguna capacidad de sistema que sería ortogonal a las requeridas por otro. Por ejemplo, al servidor de ventanas se le concedería acceso a eventos del teclado y de la pluma emitidos por el núcleo, pero no tendría permiso para acceder a la pila de operaciones del teléfono. De la misma manera, al servidor telefónico se le concedería acceso a la pila de operaciones del teléfono, pero no tendría permiso para recoger eventos del núcleo.

Como ejemplos, podemos nombrar:

EscribirDatosSistema	Permite la modificación de los datos del sistema de configuración.
ComDS	Concede acceso a todos controladores de los dispositivos de comunicaciones y de tarjetas de Ethernet.
AdminDisco	Puede llevar a cabo tareas administrativas en el disco (volver a darle formato, cambiar el nombre de una unidad...).

2.2. Capacidades expuestas al usuario: Correspondencia de los permisos en la práctica

El proceso de generar capacidades puede ser difícil. Hay que identificar en primer lugar los accesos que requieren vigilancia y, a continuación, establecer correspondencias entre esos requisitos y algo que sea significativo para un usuario. Además, más capacidades significan mayor complejidad, y suele reconocerse que la complejidad es el principal enemigo de la seguridad. Por lo tanto, una solución basada en capacidades debería procurar minimizar el número global desplegado. Las siguientes capacidades se corresponden muy ampliamente con las principales amenazas que son un acceso no autorizado a los servicios del sistema (por ejemplo, a la pila de operaciones del teléfono) y al mantenimiento de la confidencialidad/integridad de los datos del usuario.

RedTelefónica. "Puede acceder a los servicios de la red telefónica y, potencialmente, gastar dinero del usuario"

"Efectuar llamadas telefónicas".
"Enviar mensajes cortos de texto".

EscribirDatosUsuario. "Puede leer y modificar la información privada de los usuarios"

"Añadir un contacto".
"Borrar una cita".

LeerDatosUsuario. "Puede leer la información privada de los usuarios"

"Acceder a los datos de los contactos".
"Acceder a los datos de la agenda".

RedLocal. "Puede acceder a la red local" "Enviar mensajes por Bluetooth". "Establecer una conexión IR".

"Establecer una conexión USB".

Ubicación. "Puede acceder a la ubicación actual del dispositivo"

"Ubicar el dispositivo en un mapa".
"Presentar los restaurantes y el cine más cercanos".

Es necesario establecer una diferencia entre RedTelefónica y RedLocal, porque es posible transmitir información en una red sin gastar dinero (por ejemplo, una picorred Bluetooth). Este tipo de acceso puede ser un habilitador muy útil para programas de terceros, pero, pese a ello, representa una manera local de filtrar información sensible por medio de un caballo troyano, de modo que debe ser protegido con una capacidad, aunque sea RedLocal. Si el usuario la concede, RedTelefónica autorizaría a troyanos que quisiesen usar la red telefónica como vía de salida; potencialmente, eso es mucho más dañino; de ahí la tajante advertencia de su descripción. Las capacidades de raíz

y de sistema son obligatorias; si no se le conceden a un ejecutable, el usuario el dispositivo no puede decidir hacerlo. Su estricto control garantiza la integridad de la plataforma informática de confianza. Sin embargo, la forma en que los servidores verifican las capacidades expuestas al usuario o en que las interpretan puede ser completamente flexible e incluso discrecional para el usuario.

La **Figura 1** muestra procesos del momento de ejecución, anillos de capacidades del sistema y agrupaciones de capacidades visibles para el usuario.

2.3 Asignación de capacidades a un proceso

La asociación de una capacidad para el momento de ejecución con un proceso implica al cargador. En esencia, transforma las configuraciones de capacidades estáticas asociadas con las bibliotecas y los programas individuales en una capacidad de tiempo de ejecución que el núcleo mantiene y que puede ser consultada por medio de una biblioteca API de usuario en el núcleo. El cargador aplica las siguientes reglas:

Regla 1. Cuando se crea un proceso desde un programa, el cargador asigna el mismo conjunto de capacidades que las de su programa.
Regla 2. Cuando se carga una biblioteca dentro de un ejecutable, el conjunto de capacidades de la biblioteca tiene que ser mayor o igual que el conjunto de capacidades del propio ejecutable. Si no es así, la biblioteca no se carga en el ejecutable.
Regla 3. Un ejecutable puede cargar una biblioteca con capacidades superiores, pero no obtiene capacidades por hacerlo.
Regla 4. El cargador se niega a cargar ningún ejecutable que no esté en la parte cerrada de datos del sistema de ficheros reservada a la BIC.

Debe hacerse notar que:

- Las capacidades de las bibliotecas se comprueban únicamente en el momento de la carga. Más allá de eso, todo el código contenido en las bibliotecas se ejecuta libremente y se le asigna el mismo conjunto de capacidades que el programa en el que se ejecuta cuando se inician algunas llamadas de CEP.
- Para las imágenes ROM que tienen habilitada la ejecución, la herramienta de la compilación de ROM resuelve todos los símbolos realizando la misma labor que el cargador en el momento de la ejecución. Por lo tanto, la herramienta de la compilación de ROM debe imponer las mismas reglas que el cargador cuando compila una imagen ROM.

Estas reglas

- evitan que se carguen programas malignos en procesos sensibles; por ejemplo, un módulo de ampliación en un servidor del sistema fomenta la encapsulación de código sensible dentro de procesos sin derivación posible.
- Los ejemplos que siguen muestran cómo se aplican las reglas en los casos de bibliotecas cargadas de forma estática y dinámica, respectivamente.

2.3.1 Ejemplos de DLL enlazadas

El programa P.EXE está enlazado con la biblioteca L1.DLL. La biblioteca L1.DLL está enlazada con la biblioteca L0.DLL.

Caso 1:

P.EXE tiene la Cap1 y la Cap2
L1.DLL tiene la Cap1, la Cap2 y la Cap3 L0.DLL tiene la Cap1 y la Cap2
No puede crearse el proceso P. El cargador no lo aprueba porque L1.DLL no puede cargar L0.DLL, dado que L0.DLL no tiene un conjunto de capacidades mayor o igual que L1.DLL, aplicando la Regla 2.

Caso 2:

P.EXE tiene la Cap1 y la Cap2
L1.DLL tiene la Cap1, la Cap2 y la Cap3
L0.DLL tiene la Cap1, la Cap2, la Cap3 y la Cap4
Se crea el proceso P. El cargador lo consigue, y al nuevo proceso se le asignan la Cap1 y la Cap2. La capacidad del nuevo proceso se determina aplicando la Regla 1; L1.DLL no puede adquirir la capacidad Cap4 que tiene L0.DLL, y, tal como define la Regla 3, P1.EXE no puede adquirir la capacidad Cap3 que tiene L1.DLL.

2.3.2 Ejemplos de DLL cargadas dinámicamente

El programa P.EXE carga dinámicamente la biblioteca L1.DLL. Después, la biblioteca L1.DLL carga dinámicamente la biblioteca L0.DLL.

Caso 1:

P.EXE tiene la Cap1 y la Cap2
L1.DLL tiene la Cap1, la Cap2 y la Cap3 L0.DLL tiene la Cap1 y la Cap2
Se crea con éxito el proceso P y se le asignan la Cap1 y la Cap2. Cuando P solicita al cargador que cargue L1.DLL y L0.DLL, el cargador lo logra, porque P puede cargar L1.DLL y L0.DLL. La Regla 2 sí se aplica aquí, al ser el ejecutable de la carga el proceso P y no la biblioteca L1.DLL: la solicitud de carga CEP que procesa el cargador es enviada por el proceso P. El hecho de que llamada se produzca dentro de L1.DLL es aquí irrelevante. Como antes, se aplican las Reglas 1 y 3, y P no adquiere la Cap3 por el hecho de cargar L1.DLL.

Caso 2:

P.EXE tiene la Cap1 y la Cap2
L1.DLL tiene la Cap1, la Cap2 y la Cap3 L0.DLL tiene la Cap1, la Cap2 y la Cap4
Se crea con éxito el proceso P y se le asignan la Cap1 y la Cap2. Cuando P solicita al cargador que cargue L1.DLL y L0.DLL, el cargador lo logra, porque P puede cargar L1.DLL y L0.DLL. Nuevamente, sí se aplica la Regla 2, al ser P el ejecutable de la carga, no L1.DLL, mientras que la Regla 3 garantiza que P no adquiere ni la Cap3 ni la Cap4.

2.4. Protección de los nombres de los servidores del sistema

Para evitar la suplantación de servidores críticos, se ha creado una capacidad, ServProt, específica del sistema. Cuando se concede esta capacidad a un proceso, puede registrarse como “Servidor protegido” ante el núcleo con un formato específico de nombre, que empiece, por ejemplo, con “!”. El núcleo rechazará cualquier nombre de registro que empiece con “!” si el proceso llamante no tiene ServProt. Sin embargo, debe confiarse en que ninguno de los servidores de la comunidad de servidores protegidos suplante a otro servidor de esta comunidad. Se cree que este sencillo mecanismo logrará un nivel de seguridad suficientemente bueno, con la condición de que ServProt sea concedido únicamente a un conjunto pequeño de servicios. Si se comprobase que dos dominios de servidores no son suficientes, nada evitaría en el futuro añadir un nuevo dominio vinculado a una nueva capacidad y a un nuevo formato de nombre.

2.5. Aislar los servidores de sistema del SO Symbian mediante cortafuegos

2.5.1 Capacidad de duración de un permiso

Estando habilitado el modelo de capacidad de procesos mediados por el núcleo, las decisiones de las normativas específicas se delegan en los propios servidores. Cada servidor tendría que vigilarse a sí mismo y lo haría de manera diferente: algunos serían más paranoicos que otros y pueden decidir comprobar la configuración de las capacidades del cliente por acceso, o pueden negarse en redondo a permitir el acceso sin que esté activado el imprescindible bit de capacidad. Delegar las normativas de seguridad a una decisión local en los servidores representa una buena abstracción, dado que ello significa que una norma de seguridad puede ser interpretada de manera local según se desee. Además, la interpretación de la presencia del bit de capacidad podría hacerse de una de cuatro maneras; concretamente, permiso global, permiso de proceso, permiso de sesión y permiso de acceso selectivo.

Desde el punto de vista de un usuario, cada vez viene resultando más difícil identificar la duración de un proceso o de una sesión. Por ejemplo, en la mayoría de los teléfonos inteligentes, cuando el usuario accede a su agenda, no sabe si se ha creado un nuevo proceso de agenda o si se está reutilizando uno existente. De la misma manera, cuando efectúa una llamada telefónica, no tiene ni idea de si la aplicación telefónica ha creado una nueva sesión con el servidor telefónico o si reutiliza la anterior. Los inventores creen que los usuarios pueden entender solamente dos maneras de conceder capacidades: el permiso global y el permiso de acceso selectivo.

En línea con esto, habría dos alternativas posibles para la interpretación de la presencia de un bit de capacidad expuesta al usuario en los servidores conscientes de la capacidad:

Permiso global: La capacidad se concede de una vez para siempre. Si la capacidad no está presente cuando se crea el proceso, la petición falla. Se vigilará de esta manera todo acceso que requiera la capacidad del sistema.

Permiso de acción única: La capacidad se comprueba con cada llamada API sensible que se hace ala servidor. Esto significa que en cada acceso a la llamada a la API sensible, un cuadro de diálogo creado por el servidor pregunta al usuario si desea conceder esa capacidad. Sin embargo, la capacidad solo está vigente durante el transcurso de la correspondiente llamada de la API sensible.

La percepción que el usuario tiene de este tipo de permiso no debe confundirse con cuándo comprobará un servidor una capacidad requerida. Por ejemplo, a una aplicación puede concedérsele RedTelefónica como permiso global, pero el servidor telefónico comprobará la capacidad RedTelefónica en cada llamada relevante realizada por esta aplicación, dado que, en cualquier caso, el núcleo enviará el conjunto de capacidades del cliente en cada llamada CEP. Sin embargo, como al usuario no se le pedirá que adopte una decisión (conceder o no conceder RedTelefónica para esta llamada), esta comprobación de seguridad es transparente para el

usuario.

Las ventajas de este diseño son:

- Tener en cuenta inmediatamente cualquier modificación del conjunto de permiso global de proceso.
- Mantener el momento de la comprobación tan cerca del momento de uso como sea posible para evitar cualquier fallo TOCTOU (momento de la comprobación, momento del uso).
- Evitar que el servidor almacene información relativa a capacidades para un proceso.

3 Instalador de programas: Mejora de la seguridad perimetral

El instalador de programas tiene que soportar el modelo de capacidades de los procesos para determinar qué capacidades pueden concederse realmente a un trozo de código que haya de instalarse después del momento de la fabricación. Hay muchas maneras de hacerlo, y la presente invención no impone ninguna. En las siguientes secciones, exploramos diferentes escenarios basados en la existencia de una infraestructura de clave pública para ilustrar cómo alguien puede escoger hacer y qué asuntos deberían considerarse.

El instalador de programas puede soportar tres escenarios principales:

- La [des]instalación/revocación de aplicaciones de confianza (firmadas)
- La [des]instalación/revocación de aplicaciones que no son de confianza (firmadas)
- La [des]instalación/revocación de aplicaciones no firmadas

Aquí, la distinción de confianza es una función de la presencia de un certificado raíz en la cadena de firmas que se corresponde a uno de los certificados raíz almacenados en el dispositivo. En las subsecciones siguientes examinamos estas alternativas desde la perspectiva de un usuario.

3.1 Instalación de programas de confianza (firmados)

En este caso, las aplicaciones cliente autorizadas serían revisadas por una autoridad reconocida. Entonces los programas estarían firmados, junto con una lista de las capacidades que requerían, en paquetes de instalación. Las capacidades podrían estar firmadas para capacidades hipotéticas expuestas al sistema y al usuario. La cadena de certificados incluidos con los programas está terminada en una AC raíz conocida por el instalador del programa, en cuyo caso el usuario puede instalar el programa con seguridad por el conocimiento de que la AC raíz responde del mismo. En el momento de la instalación, el programa se instalaría en el dispositivo sin ninguna respuesta del usuario, aun- que en algunos dispositivos sería posible examinar los detalles del certificado de la firma. Obsérvese que cualquier programa de terceros que buscase la capacidad de sistema sería obligado a seguir esta ruta.

3.2 Instalación de programas que no son de confianza (firmados)

En este caso, el programa de terceros no satisfaría una verificación contra ninguno de los certificados raíz. El programa está firmado por una AC raíz desconocida, en cuyo caso al usuario se le presentaría una advertencia, pero podría simplemente seguir adelante e instalar el programa de todos modos tras ver el contenido del certificado. El instalador del programa presentaría un diálogo de usuario en el momento de la instalación, lo que ofrecería al usuario la opción de instalar la aplicación más una lista de las capacidades que la aplicación deseara solicitar. Las capacidades que pudieran ser concedidas siguiendo esta ruta estarían en la gama de capacidades expuestas al usuario. En otras palabras, no sería posible que los usuarios concediesen la capacidad de sistema a programas firmados que no satisficieran la verificación contra uno de los certificados raíz en el dispositivo.

3.3 Instalación de programas no firmados

Aquí el escenario sería similar a 3.2. Los terceros tendrían que solicitar capacidades con su paquete de instalación. Nuevamente, por medio de esta ruta únicamente estarían disponibles las capacidades expuestas al usuario: no sería posible que los usuarios concediesen la capacidad de sistema a programas no firmados.

La **Figura 2** ilustra un proceso posible de instalación cuando se desea dejar que el usuario adopte decisiones discrecionales en cuanto a la normativa a aplicar ya sea a una aplicación no firmada o a una aplicación firmada cuyo certificado raíz de firma no cuadra con ninguno de los certificados raíz del dispositivo.

3.4 Panel de control para revocar capacidades

En el caso de que se desee dejar que el usuario adopte decisiones discrecionales en cuanto a normas de seguridad después del momento de la instalación, se requeriría un diálogo de un panel de control del instalador de programas para proporcionar acceso a las configuraciones de capacidades otorgadas en ese momento a diversas aplicaciones de terceros instaladas. El usuario podría ver y, opcionalmente, conceder o revocar las capacidades expuestas al usuario para el código de terceros instalado. Además, (en caso necesario) sería posible ver un certificado de firmas correspondiente del código. La siguiente tabla ilustra el aspecto que podría tener el diálogo del panel de control.

Instalado	RedTelefonica	LeerDatosUsuario	EscribirDatosUsuario	RedLocal	Certificado
Bueno.exe	No concedido	Concedido	No concedido	Concedido	Verificación válida. Hacer clic para ver el certificado de firma raíz
Malo.exe	No concedido	No concedido	No concedido	No concedido	No firmado

REIVINDICACIONES

1. Un dispositivo inalámbrico móvil seguro para un solo usuario en el que está instalado código ejecutable nativo, incluyendo el dispositivo:

una pluralidad de recursos protegidos; una pluralidad de servidores; y
una base informática de confianza que tiene un núcleo; en el que
el acceso a dicho recurso protegido es proporcionado por un correspondiente servidor;
al código ejecutable nativo se le asigna un conjunto de capacidades que definen un/los recurso(s) protegido(s) en el dispositivo a los que puede acceder el código ejecutable nativo;
dichos servidores correspondientes están dispuestos para que vigilen el acceso a dichos recursos protegidos en base a las capacidades asignadas al código ejecutable nativo;
las capacidades están almacenadas en una ubicación que es accesible únicamente a la base informática de confianza; y el núcleo está dispuesto, para cada comunicación cliente-servidor, para pasar las capacidades del cliente a dicho servidor.

2. El dispositivo de la Reivindicación 1 en el que una capacidad autoriza el acceso a una API (o a una gama de API), a un fichero específicos o a otro(s) recurso(s) protegido(s) del sistema operativo, y donde las capacidades del cliente se comprueban, en dichos servidores correspondientes, en cada llamada relevante de un servidor realizada por un cliente.

3. El dispositivo de la Reivindicación 1 en el que, antes del momento de la instalación, el código ejecutable ya contiene las capacidades que se le han concedido.

4. El dispositivo de la Reivindicación 3 en el que el código ejecutable está almacenado de forma permanente en una ubicación de la memoria del dispositivo que es accesible únicamente a la base informática de confianza para mantener la integridad de las capacidades asignadas al código ejecutable.

5. El dispositivo de la Reivindicación 4 en el que un cargador se niega a cargar ejecutables no almacenados de forma permanente en una ubicación de la memoria del dispositivo que sea accesible únicamente a la base informática de confianza, para mantener la integridad del sistema.

6. El dispositivo de la Reivindicación 1 en el que hay capacidades del sistema que son obligatorias y controlan el acceso al sistema de ficheros y al sistema de datos del núcleo del sistema, y donde las capacidades del cliente se comprueban, en dichos servidores correspondientes, en cada llamada relevante de un servidor realizada por un cliente.

7. El dispositivo de la Reivindicación 1 en el que, para que una biblioteca sea cargada en un ejecutable, el cargador verifica que a la biblioteca se le haya asignado un superconjunto de las capacidades concedidas al ejecutable que hace la petición.

8. El dispositivo de la Reivindicación 1 en el que se selecciona una capacidad entre una lista de capacidades que comprenden capacidades raíz, capacidades del sistema y capacidades expuestas a los usuarios.

9. El dispositivo de la Reivindicación 8 en el que las capacidades expuestas a los usuarios permiten las siguientes funciones:

(a) Poder gastar dinero de los abonados usando la red telefónica
(b) Poder leer la información privada de los usuarios
(c) Poder modificar la información privada de los usuarios
(d) Poder enviar información a una red local sin gastar dinero de los abonados
(e) Poder conocer la ubicación del dispositivo.

10. El dispositivo de la Reivindicación 1 en el que el servidor puede ser el núcleo, y donde las capacidades del cliente se comprueban, en dichos servidores correspondientes, en cada llamada relevante de un servidor realizada por un cliente.

11. Un procedimiento para habilitar que un código ejecutable nativo, instalado en un dispositivo inalámbrico móvil para un único usuario, acceda a recursos protegidos en el dispositivo, en el que:

el dispositivo incluye: una pluralidad de dichos recursos protegidos; una pluralidad de servidores; y una base informática de confianza que tiene un núcleo;
el acceso a cada uno de dichos recursos protegidos es proporcionado por un correspondiente servidor;
al código ejecutable nativo se le asigna un conjunto de capacidades que definen un/los recurso(s) protegido(s) en el dispositivo a los que puede acceder el código ejecutable nativo; y
las capacidades se almacenan en una ubicación que es accesible únicamente a la base informática de confianza;
comprendiendo el procedimiento las etapas de:

(a) vigilar el acceso a dichos recursos protegidos en dichos servidores correspondientes en base a las capacidades asignadas al código ejecutable nativo; y
(b) para cada comunicación cliente-servidor, el núcleo pasa las capacidades del cliente a dicho servidor;
(c) comprobar, en dichos servidores correspondientes, las capacidades del cliente en cada llamada relevante de un servidor realizada por un cliente.

Figura 1

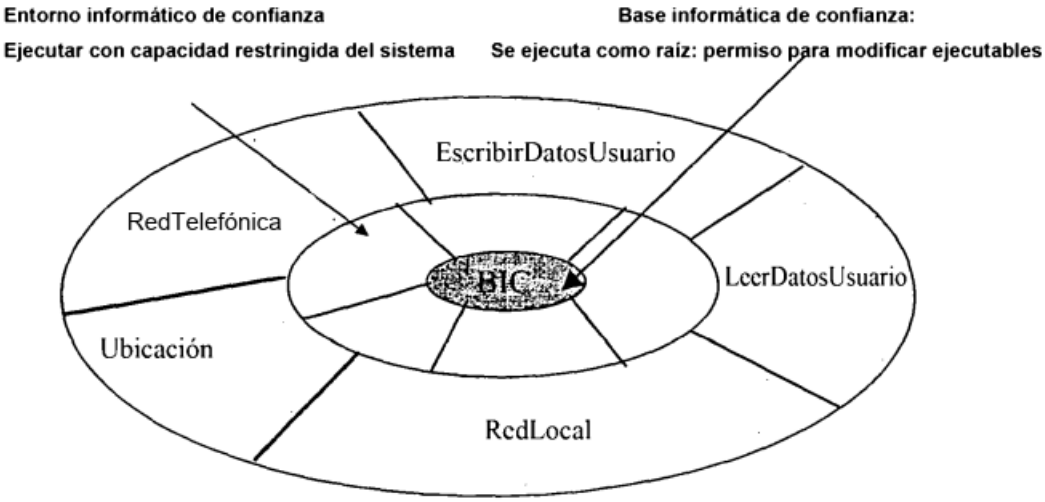
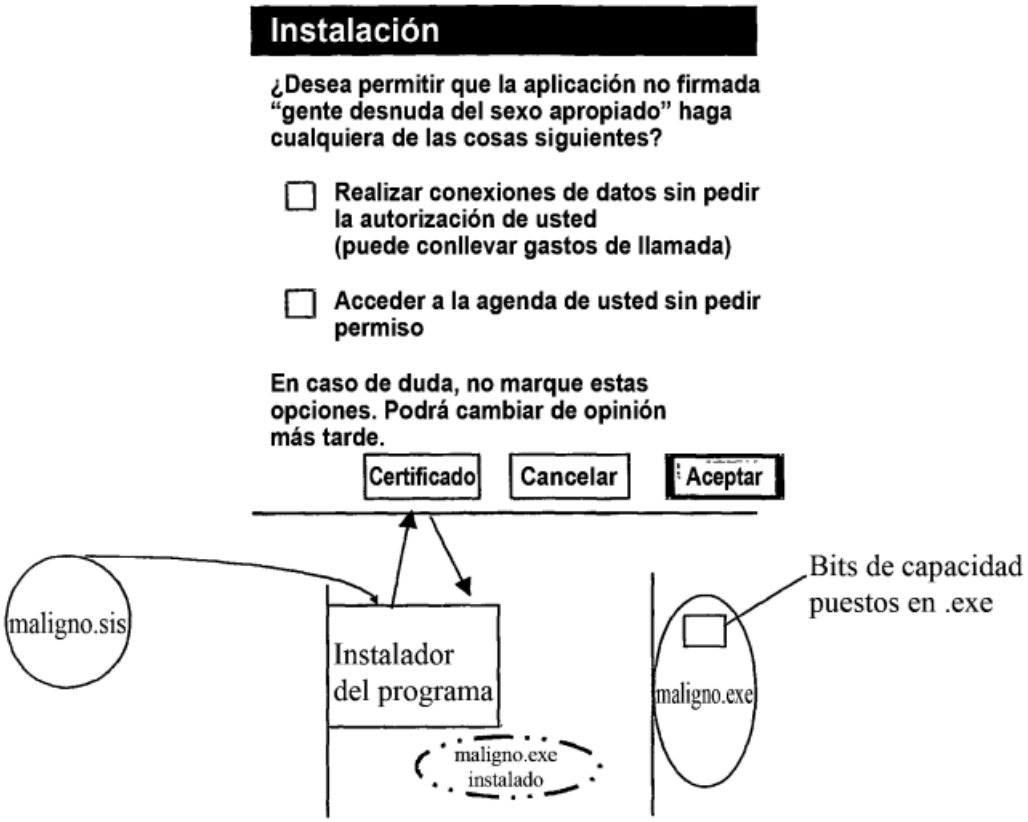

Entorno informático de confianza
Ejecutar con capacidad restringida del sistema
Base informática de confianza:
Se ejecuta como raíz: permiso para modificar ejecutables
EscribirDatosUsuario
RedTelefónica
BIC
LeerDatosUsuario
Ubicación
RedLocal

Figura 2


Instalación
¿Desea permitir que la aplicación no firmada "gente desnuda del sexo apropiado" haga cualquiera de las cosas siguientes?
Realizar conexiones de datos sin pedir la autorización de usted (puede conllevar gastos de llamada)
Acceder a la agenda de usted sin pedir permiso
En caso de duda, no marque estas opciones. Podrá cambiar de opinión más tarde.
Certificado
Cancelar
Aceptar
maligno.sis
Instalador del programa
maligno.exe instalado
maligno.exe
Bits de capacidad puestos en .exe